La finalidad de esta práctica volcar datos desde Mysql a HDFS y viceversa con Apache Sqoop.

Vete haciendo capturas de pantalla de todos los pasos que vayas dando así como su resultado, acompañándolas de comentarios descriptivos de los mismos.

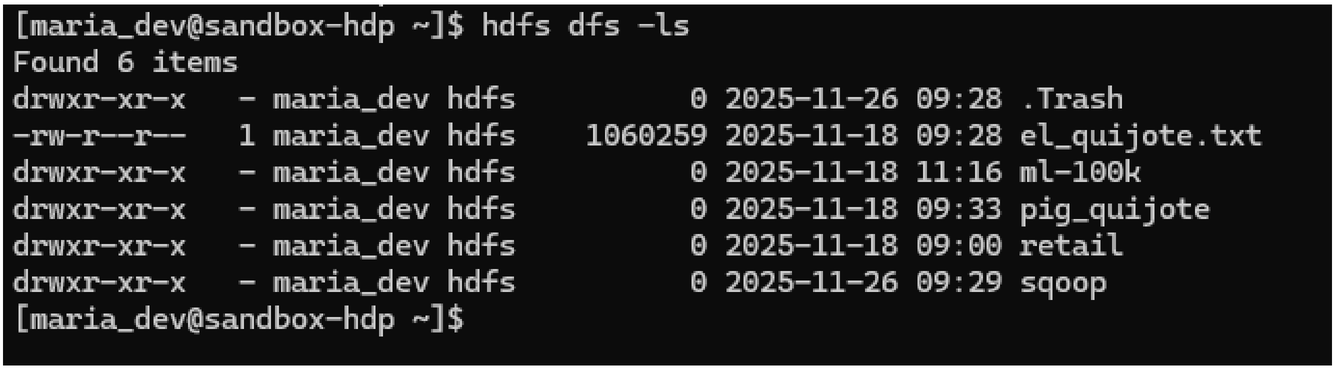
APARTADO A

INTRODUCCIÓN

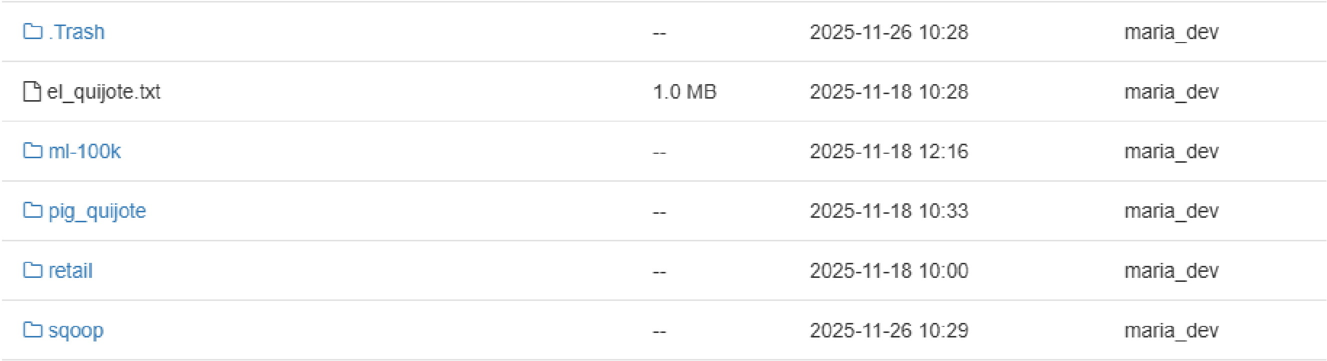
Realizaremos la práctica utilizando el servidor de MySql que hemos instalado en una máquina EC2 de AWS en la práctica anterior.

CONTENIDO

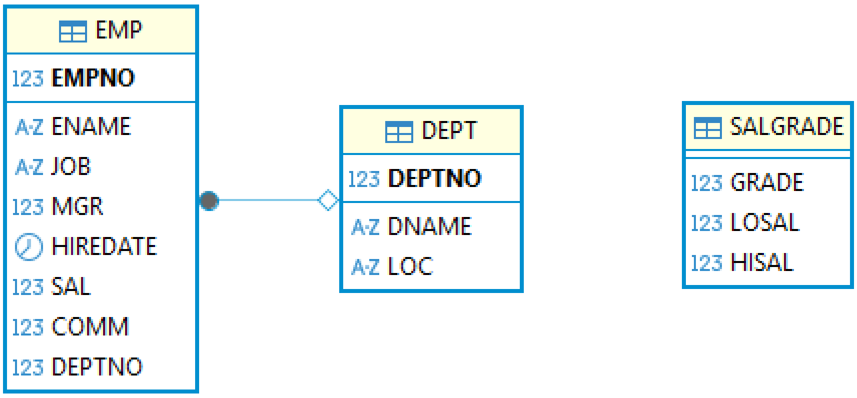
1.- En la carpeta del usuario `maria_dev` en HDFS crea una subcarpeta llamada `sqoop` donde guardaremos los archivos de esta práctica.



```
[maria_dev@sandbox-hdp ~]$ hdfs dfs -ls
Found 6 items
drwxr-xr-x   - maria_dev hdfs          0 2025-11-26 09:28 .Trash
-rw-r--r--   1 maria_dev hdfs    1060259 2025-11-18 09:28 el_quijote.txt
drwxr-xr-x   - maria_dev hdfs          0 2025-11-18 11:16 ml-100k
drwxr-xr-x   - maria_dev hdfs          0 2025-11-18 09:33 pig_quijote
drwxr-xr-x   - maria_dev hdfs          0 2025-11-18 09:00 retail
drwxr-xr-x   - maria_dev hdfs          0 2025-11-26 09:29 sqoop
[maria_dev@sandbox-hdp ~]$
```



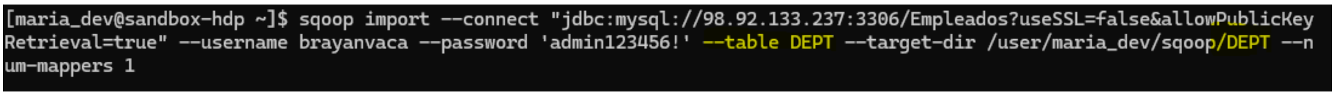
🗀 .Trash	--	2025-11-26 10:28	maria_dev
🗋 el_quijote.txt	1.0 MB	2025-11-18 10:28	maria_dev
🗀 ml-100k	--	2025-11-18 12:16	maria_dev
🗀 pig_quijote	--	2025-11-18 10:33	maria_dev
🗀 retail	--	2025-11-18 10:00	maria_dev
🗀 sqoop	--	2025-11-26 10:29	maria_dev



2.- Importa con Sqoop las tres tablas que creamos en MySql en la práctica anterior.

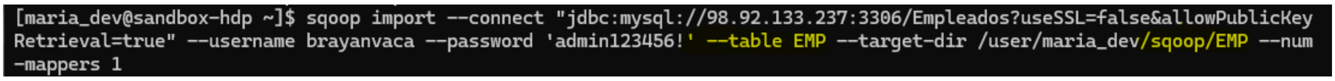
```
sqoop import \
  --connect "jdbc:mysql://98.92.133.237:3306/Empleados?useSSL=false&allowPublicKeyRetrieval=true" \
  --username brayanvaca \
  --password 'admin123456!' \
  --table DEPT \
  --target-dir /user/maria_dev/sqoop/DEPT \
  --num-mappers 1
```

Para importar la tabla DEPT



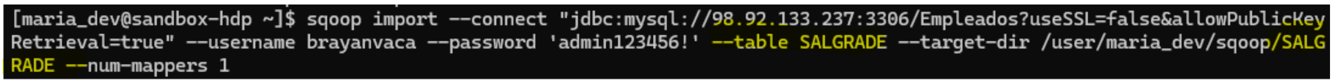
```
[maria_dev@sandbox-hdp ~]$ sqoop import --connect "jdbc:mysql://98.92.133.237:3306/Empleados?useSSL=false&allowPublicKey
Retrieval=true" --username brayanvaca --password 'admin123456!' --table DEPT --target-dir /user/maria_dev/sqoop/DEPT --n
um-mappers 1
```

Para importar la tabla EMP



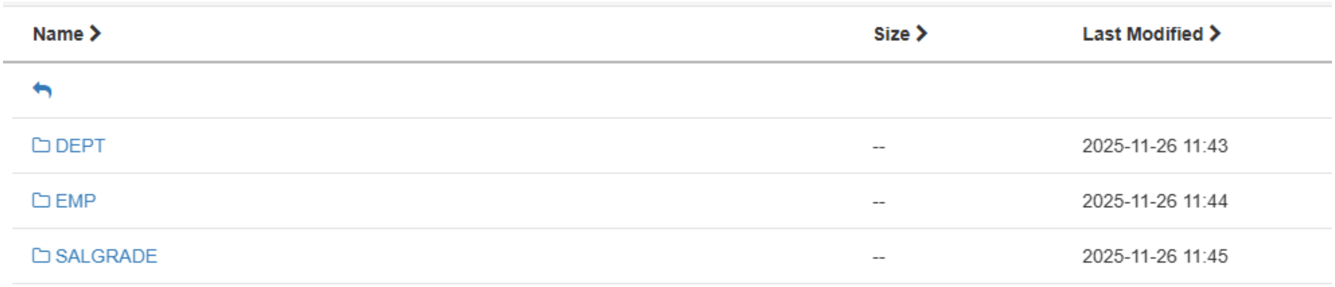
```
[maria_dev@sandbox-hdp ~]$ sqoop import --connect "jdbc:mysql://98.92.133.237:3306/Empleados?useSSL=false&allowPublicKey
Retrieval=true" --username brayanvaca --password 'admin123456!' --table EMP --target-dir /user/maria_dev/sqoop/EMP --num
-mappers 1
```

Para importar la tabla SALGRADE



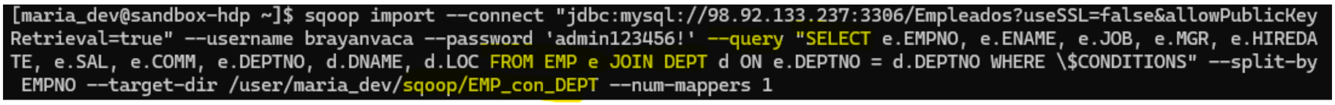
```
[maria_dev@sandbox-hdp ~]$ sqoop import --connect "jdbc:mysql://98.92.133.237:3306/Empleados?useSSL=false&allowPublicKey
Retrieval=true" --username brayanvaca --password 'admin123456!' --table SALGRADE --target-dir /user/maria_dev/sqoop/SALG
RADE --num-mappers 1
```

Ya las tenemos en HDFS

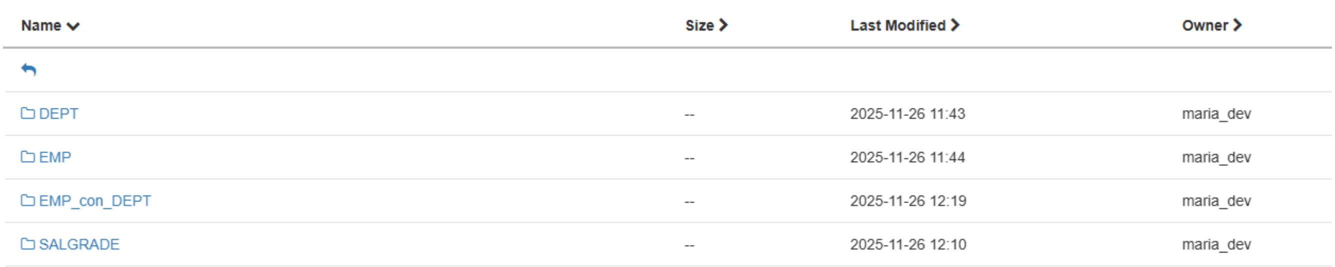


Name ❯	Size ❯	Last Modified ❯
↩		
🗀 DEPT	--	2025-11-26 11:43
🗀 EMP	--	2025-11-26 11:44
🗀 SALGRADE	--	2025-11-26 11:45

3.- Importa, dejándola en un solo archivo, todos los datos de los empleados adjuntándoles a cada uno toda la información de su departamento.



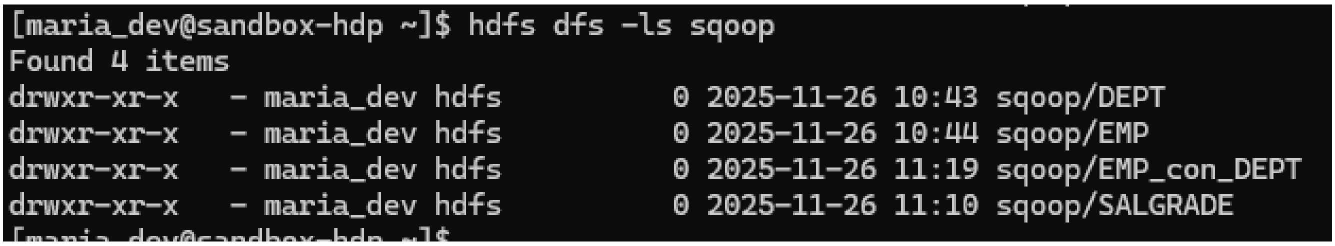
```
[maria_dev@sandbox-hdp ~]$ sqoop import --connect "jdbc:mysql://98.92.133.237:3306/Empleados?useSSL=false&allowPublicKey
Retrieval=true" --username brayanvaca --password 'admin123456!' --query "SELECT e.EMPNO, e.ENAME, e.JOB, e.MGR, e.HIREDA
TE, e.SAL, e.COMM, e.DEPTNO, d.DNAME, d.LOC FROM EMP e JOIN DEPT d ON e.DEPTNO = d.DEPTNO WHERE \$CONDITIONS" --split-by
 EMPNO --target-dir /user/maria_dev/sqoop/EMP_con_DEPT --num-mappers 1
```



Name	Size	Last Modified	Owner
↩			
DEPT	--	2025-11-26 11:43	maria_dev
EMP	--	2025-11-26 11:44	maria_dev
EMP_con_DEPT	--	2025-11-26 12:19	maria_dev
SALGRADE	--	2025-11-26 12:10	maria_dev

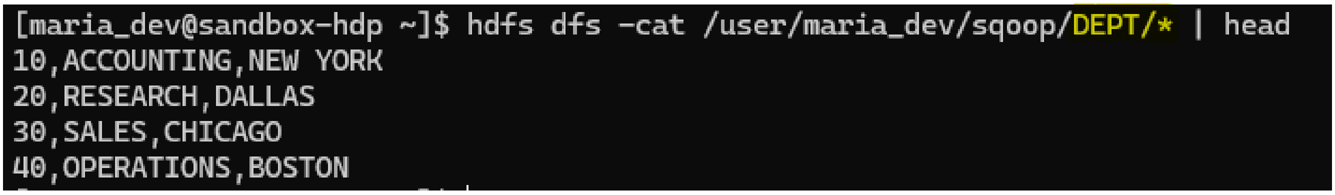
4. Muestra en HDFS la ubicación y contenido de los ficheros resultantes.

Ubicación de todas las tablas maria_dev/sqoop



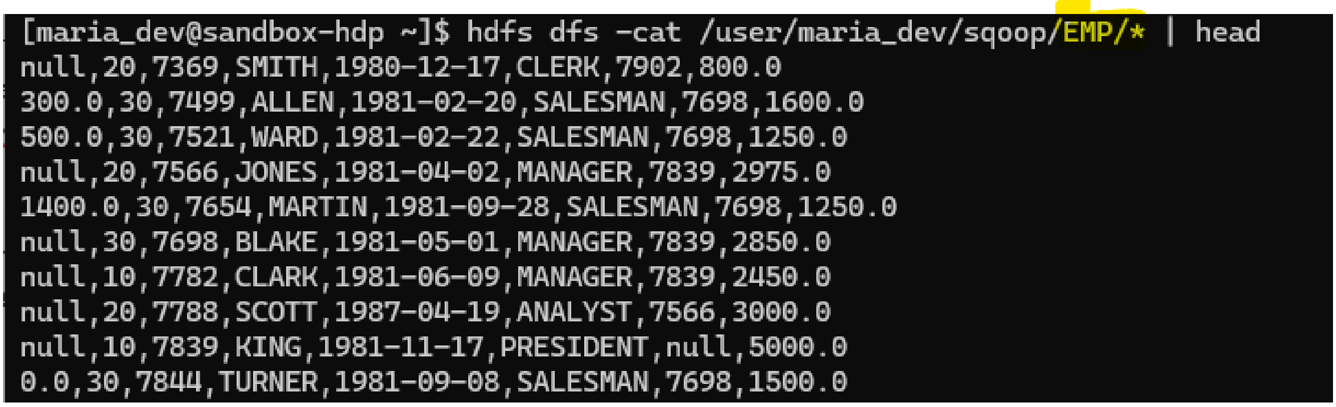
```
[maria_dev@sandbox-hdp ~]$ hdfs dfs -ls sqoop
Found 4 items
drwxr-xr-x   - maria_dev hdfs          0 2025-11-26 10:43 sqoop/DEPT
drwxr-xr-x   - maria_dev hdfs          0 2025-11-26 10:44 sqoop/EMP
drwxr-xr-x   - maria_dev hdfs          0 2025-11-26 11:19 sqoop/EMP_con_DEPT
drwxr-xr-x   - maria_dev hdfs          0 2025-11-26 11:10 sqoop/SALGRADE
[maria_dev@sandbox-hdp ~]$
```

Contenido de la tabla DEPT



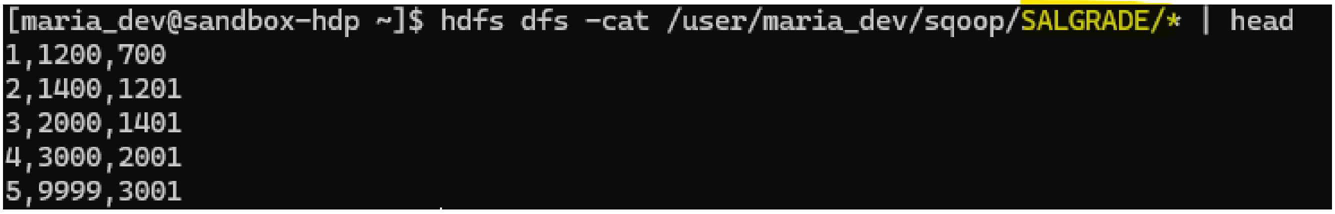
```
[maria_dev@sandbox-hdp ~]$ hdfs dfs -cat /user/maria_dev/sqoop/DEPT/* | head
10,ACCOUNTING,NEW YORK
20,RESEARCH,DALLAS
30,SALES,CHICAGO
40,OPERATIONS,BOSTON
```

Contenido de la tabla EMP



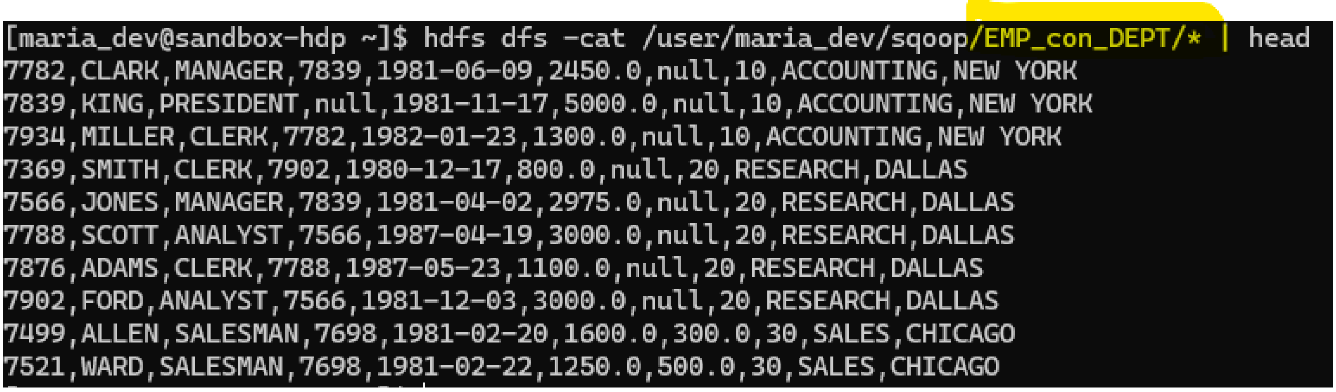
```
[maria_dev@sandbox-hdp ~]$ hdfs dfs -cat /user/maria_dev/sqoop/EMP/* | head
null,20,7369,SMITH,1980-12-17,CLERK,7902,800.0
300.0,30,7499,ALLEN,1981-02-20,SALESMAN,7698,1600.0
500.0,30,7521,WARD,1981-02-22,SALESMAN,7698,1250.0
null,20,7566,JONES,1981-04-02,MANAGER,7839,2975.0
1400.0,30,7654,MARTIN,1981-09-28,SALESMAN,7698,1250.0
null,30,7698,BLAKE,1981-05-01,MANAGER,7839,2850.0
null,10,7782,CLARK,1981-06-09,MANAGER,7839,2450.0
null,20,7788,SCOTT,1987-04-19,ANALYST,7566,3000.0
null,10,7839,KING,1981-11-17,PRESIDENT,null,5000.0
0.0,30,7844,TURNER,1981-09-08,SALESMAN,7698,1500.0
```

Contenido de la tabla SALGRADE



```
[maria_dev@sandbox-hdp ~]$ hdfs dfs -cat /user/maria_dev/sqoop/SALGRADE/* | head
1,1200,700
2,1400,1201
3,2000,1401
4,3000,2001
5,9999,3001
```

Contenido de la tabla de Todos los empleados con la información de su departamento



```
[maria_dev@sandbox-hdp ~]$ hdfs dfs -cat /user/maria_dev/sqoop/EMP_con_DEPT/* | head
7782,CLARK,MANAGER,7839,1981-06-09,2450.0,null,10,ACCOUNTING,NEW YORK
7839,KING,PRESIDENT,null,1981-11-17,5000.0,null,10,ACCOUNTING,NEW YORK
7934,MILLER,CLERK,7782,1982-01-23,1300.0,null,10,ACCOUNTING,NEW YORK
7369,SMITH,CLERK,7902,1980-12-17,800.0,null,20,RESEARCH,DALLAS
7566,JONES,MANAGER,7839,1981-04-02,2975.0,null,20,RESEARCH,DALLAS
7788,SCOTT,ANALYST,7566,1987-04-19,3000.0,null,20,RESEARCH,DALLAS
7876,ADAMS,CLERK,7788,1987-05-23,1100.0,null,20,RESEARCH,DALLAS
7902,FORD,ANALYST,7566,1981-12-03,3000.0,null,20,RESEARCH,DALLAS
7499,ALLEN,SALESMAN,7698,1981-02-20,1600.0,300.0,30,SALES,CHICAGO
7521,WARD,SALESMAN,7698,1981-02-22,1250.0,500.0,30,SALES,CHICAGO
```

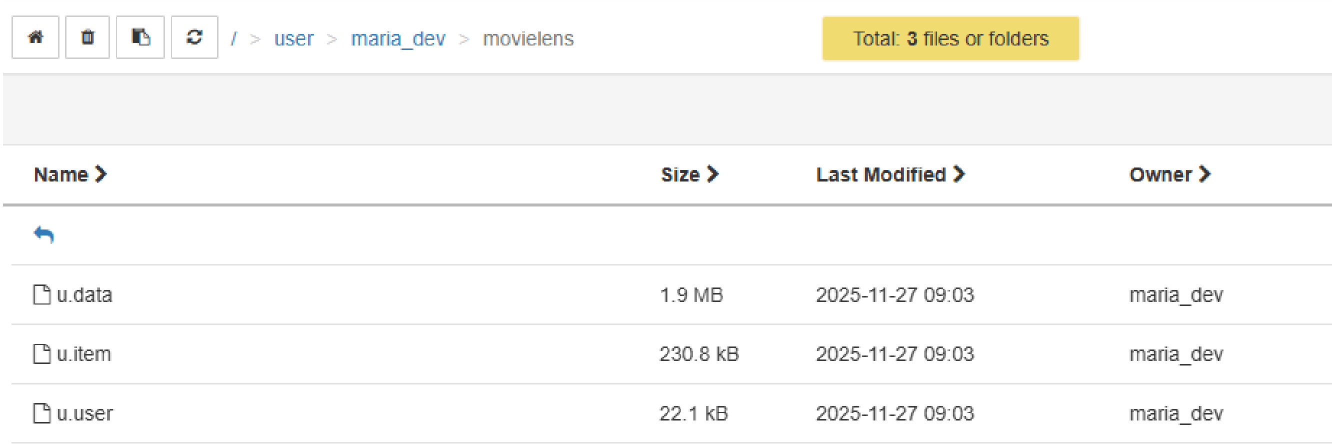
APARTADO B

INTRODUCCIÓN

En este práctica trabajaremos con los ficheros del *dataset* de Movielens de la práctica 3.

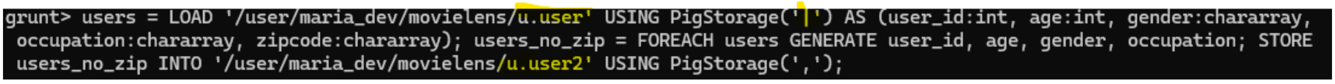
CONTENIDO

1.- En la carpeta del usuario `maria_dev` en HDFS crea una subcarpeta llamada `movielens` donde guardaremos los archivos `u.data`, `u.user` y `u.item`.

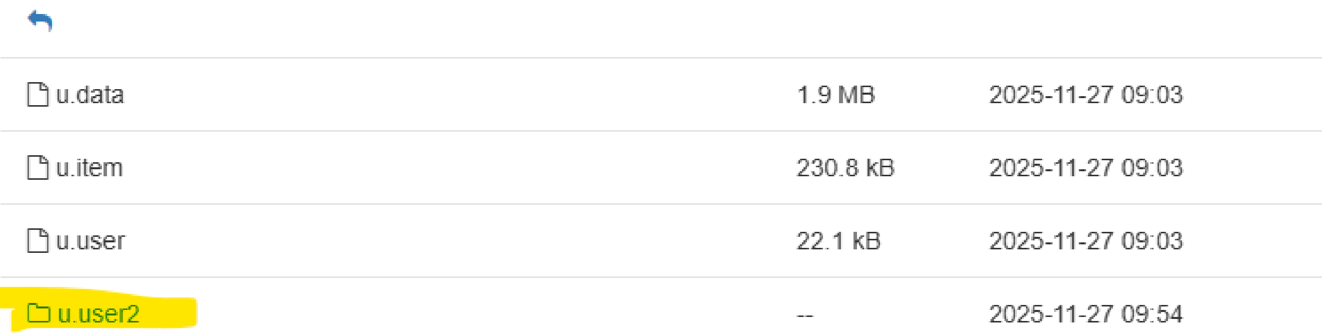


Name	Size	Last Modified	Owner
u.data	1.9 MB	2025-11-27 09:03	maria_dev
u.item	230.8 kB	2025-11-27 09:03	maria_dev
u.user	22.1 kB	2025-11-27 09:03	maria_dev

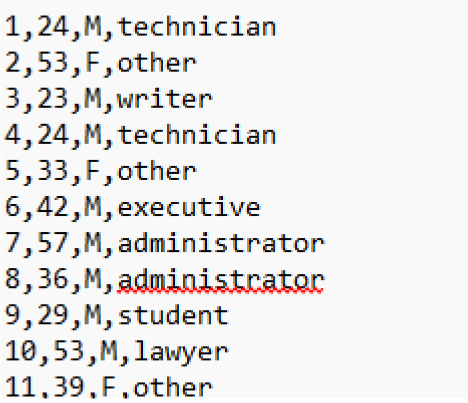
2.- Utilizando PIG, al archivo `u.user` quítale la última columna con el código postal. Guarda el resultado en el archivo `u.user2`.



```
grunt> users = LOAD '/user/maria_dev/movielens/u.user' USING PigStorage('|') AS (user_id:int, age:int, gender:chararray,
 occupation:chararray, zipcode:chararray); users_no_zip = FOREACH users GENERATE user_id, age, gender, occupation; STORE
 users_no_zip INTO '/user/maria_dev/movielens/u.user2' USING PigStorage(',');
```

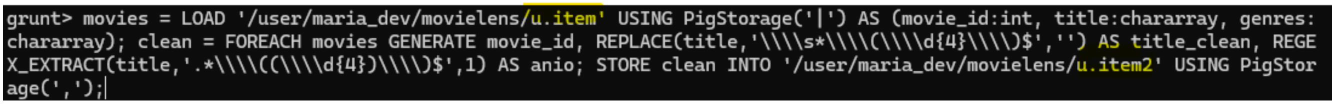


u.data	1.9 MB	2025-11-27 09:03
u.item	230.8 kB	2025-11-27 09:03
u.user	22.1 kB	2025-11-27 09:03
u.user2	--	2025-11-27 09:54

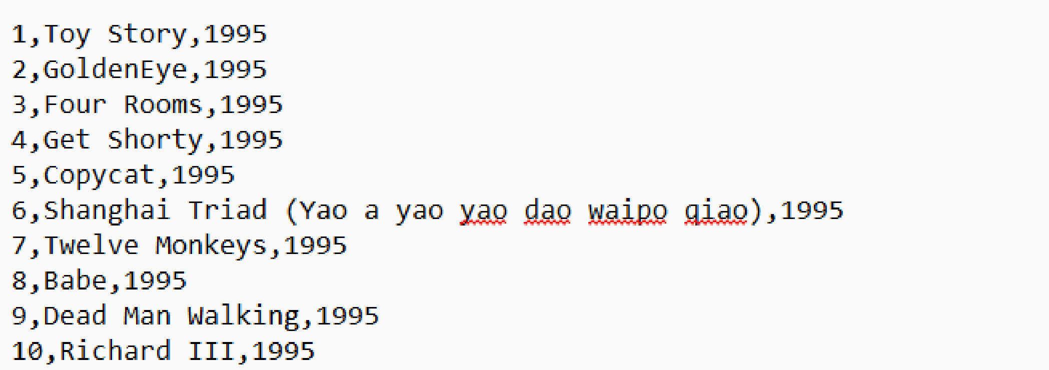


```
1,24,M,technician
2,53,F,other
3,23,M,writer
4,24,M,technician
5,33,F,other
6,42,M,executive
7,57,M,administrator
8,36,M,administrator
9,29,M,student
10,53,M,lawyer
11,39,F,other
```

3.- Utilizando PIG, del archivo `u.item` quédate solamente con las dos primeras columnas (id y título). Posteriormente de la columna título extrae el año y guárdala en una nueva columna (`anio`). En la columna título ha de quedar exclusivamente el título sin el año. Guarda el resultado en un archivo llamado `u.item2`



```
grunt> movies = LOAD '/user/maria_dev/movielens/u.item' USING PigStorage('|') AS (movie_id:int, title:chararray, genres:
chararray); clean = FOREACH movies GENERATE movie_id, REPLACE(title,'\\\\s*\\\\(\\\\d{4}\\\\)$','') AS title_clean, REGE
X_EXTRACT(title,'.*\\\\((\\\\d{4})\\\\)$',1) AS anio; STORE clean INTO '/user/maria_dev/movielens/u.item2' USING PigStor
age(',');
```



```
1,Toy Story,1995
2,GoldenEye,1995
3,Four Rooms,1995
4,Get Shorty,1995
5,Copycat,1995
6,Shanghai Triad (Yao a yao yao dao waipo qiao),1995
7,Twelve Monkeys,1995
8,Babe,1995
9,Dead Man Walking,1995
10,Richard III,1995
```

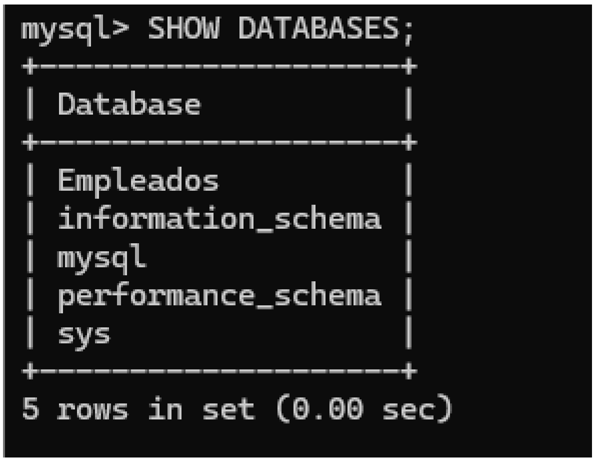
APARTADO C

INTRODUCCIÓN

Realizaremos la práctica utilizando el servidor de MySql que hemos instalado en una máquina EC2 de AWS en la práctica anterior.

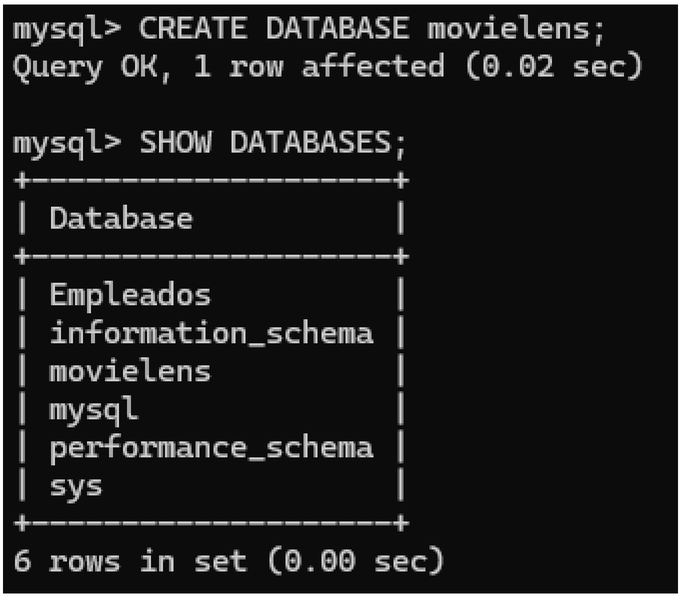
CONTENIDO

Bases de datos actuales



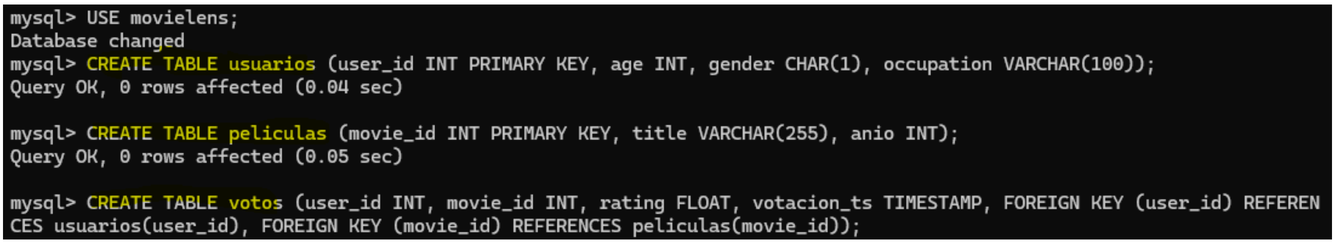
```
mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| Empleados          |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)
```

1.- Crea en tu servidor MySql de AWS una nueva base de datos llamada `movielens`.

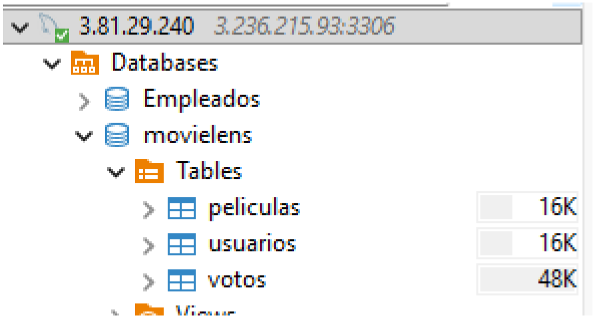


```
mysql> CREATE DATABASE movielens;
Query OK, 1 row affected (0.02 sec)

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| Empleados          |
| information_schema |
| movielens          |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0.00 sec)
```

2.- En dicha base de datos crea tres tablas (`usuarios`, `votos` y `peliculas`) con la estructura adecuada para almacenar los ficheros `u.data`, `u.user2` y `u.item2`. Utiliza sentencias `CREATE TABLE`. Crea los índices y relaciones entre las tres tablas.

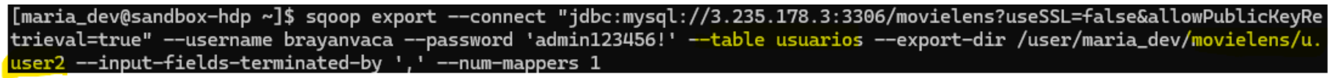


```
mysql> USE movielens;
Database changed
mysql> CREATE TABLE usuarios (user_id INT PRIMARY KEY, age INT, gender CHAR(1), occupation VARCHAR(100));
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE peliculas (movie_id INT PRIMARY KEY, title VARCHAR(255), anio INT);
Query OK, 0 rows affected (0.05 sec)

mysql> CREATE TABLE votos (user_id INT, movie_id INT, rating FLOAT, votacion_ts TIMESTAMP, FOREIGN KEY (user_id) REFEREN
CES usuarios(user_id), FOREIGN KEY (movie_id) REFERENCES peliculas(movie_id));
```

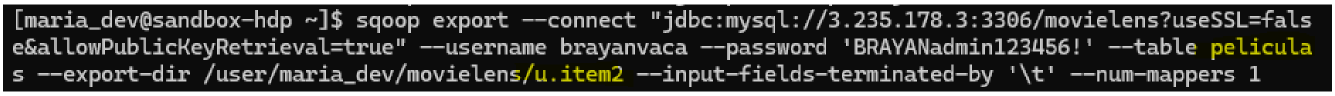


```
3.81.29.240   3.236.215.93:3306
  Databases
    Empleados
    movielens
      Tables
        peliculas    16K
        usuarios     16K
        votos        48K
      Views
```

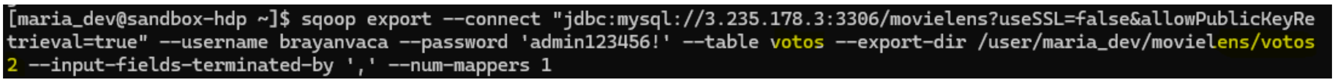
3.- Utilizando SQOOP, exporta los tres ficheros de HDFS a sus tablas.



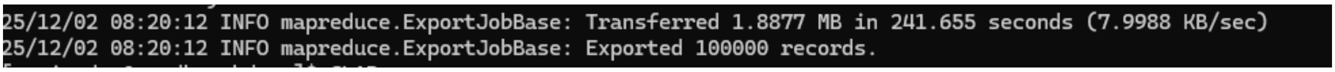
```
[maria_dev@sandbox-hdp ~]$ sqoop export --connect "jdbc:mysql://3.235.178.3:3306/movielens?useSSL=false&allowPublicKeyRe
trieval=true" --username brayanvaca --password 'admin123456!' --table usuarios --export-dir /user/maria_dev/movielens/u.
user2 --input-fields-terminated-by ',' --num-mappers 1
```



```
[maria_dev@sandbox-hdp ~]$ sqoop export --connect "jdbc:mysql://3.235.178.3:3306/movielens?useSSL=fals
e&allowPublicKeyRetrieval=true" --username brayanvaca --password 'BRAYANadmin123456!' --table pelicula
s --export-dir /user/maria_dev/movielens/u.item2 --input-fields-terminated-by '\t' --num-mappers 1
```

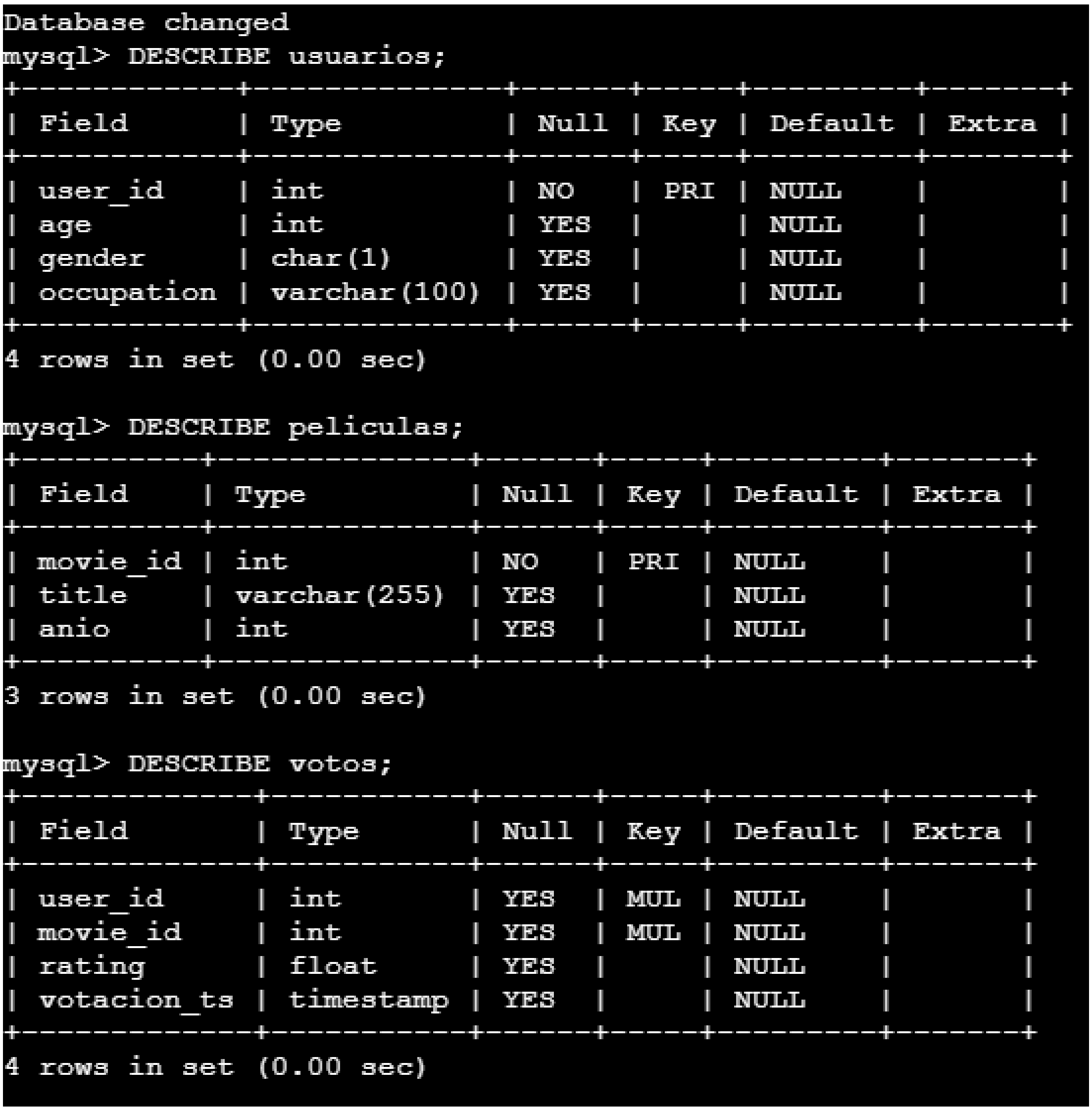


```
[maria_dev@sandbox-hdp ~]$ sqoop export --connect "jdbc:mysql://3.235.178.3:3306/movielens?useSSL=false&allowPublicKeyRe
trieval=true" --username brayanvaca --password 'admin123456!' --table votos --export-dir /user/maria_dev/movielens/votos
2 --input-fields-terminated-by ',' --num-mappers 1
```

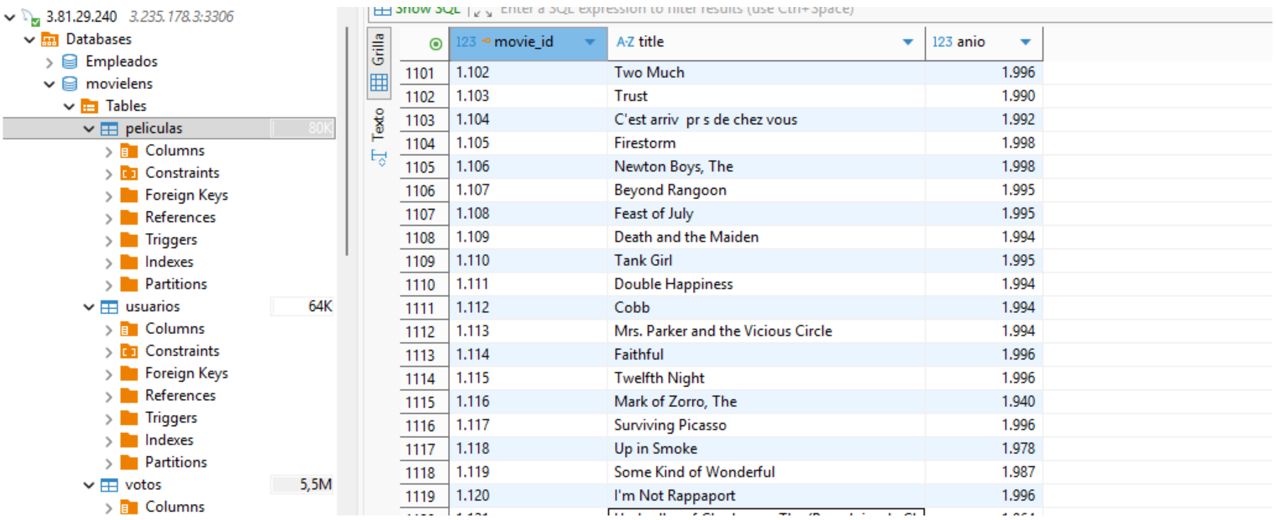


```
25/12/02 08:20:12 INFO mapreduce.ExportJobBase: Transferred 1.8877 MB in 241.655 seconds (7.9988 KB/sec)
25/12/02 08:20:12 INFO mapreduce.ExportJobBase: Exported 100000 records.
```

4.- Comprueba con sentencias SELECT que los ficheros se importaron correctamente.



```
Database changed
mysql> DESCRIBE usuarios;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| user_id    | int          | NO   | PRI | NULL    |       |
| age        | int          | YES  |     | NULL    |       |
| gender     | char(1)      | YES  |     | NULL    |       |
| occupation | varchar(100) | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> DESCRIBE peliculas;
+----------+--------------+------+-----+---------+-------+
| Field    | Type         | Null | Key | Default | Extra |
+----------+--------------+------+-----+---------+-------+
| movie_id | int          | NO   | PRI | NULL    |       |
| title    | varchar(255) | YES  |     | NULL    |       |
| anio     | int          | YES  |     | NULL    |       |
+----------+--------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> DESCRIBE votos;
+-------------+-----------+------+-----+---------+-------+
| Field       | Type      | Null | Key | Default | Extra |
+-------------+-----------+------+-----+---------+-------+
| user_id     | int       | YES  | MUL | NULL    |       |
| movie_id    | int       | YES  | MUL | NULL    |       |
| rating      | float     | YES  |     | NULL    |       |
| votacion_ts | timestamp | YES  |     | NULL    |       |
+-------------+-----------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```



	123 movie_id	A-Z title	123 anio
1101	1.102	Two Much	1.996
1102	1.103	Trust	1.990
1103	1.104	C'est arriv pr s de chez vous	1.992
1104	1.105	Firestorm	1.998
1105	1.106	Newton Boys, The	1.998
1106	1.107	Beyond Rangoon	1.995
1107	1.108	Feast of July	1.995
1108	1.109	Death and the Maiden	1.994
1109	1.110	Tank Girl	1.995
1110	1.111	Double Happiness	1.994
1111	1.112	Cobb	1.994
1112	1.113	Mrs. Parker and the Vicious Circle	1.994
1113	1.114	Faithful	1.996
1114	1.115	Twelfth Night	1.996
1115	1.116	Mark of Zorro, The	1.940
1116	1.117	Surviving Picasso	1.996
1117	1.118	Up in Smoke	1.978
1118	1.119	Some Kind of Wonderful	1.987
1119	1.120	I'm Not Rappaport	1.996

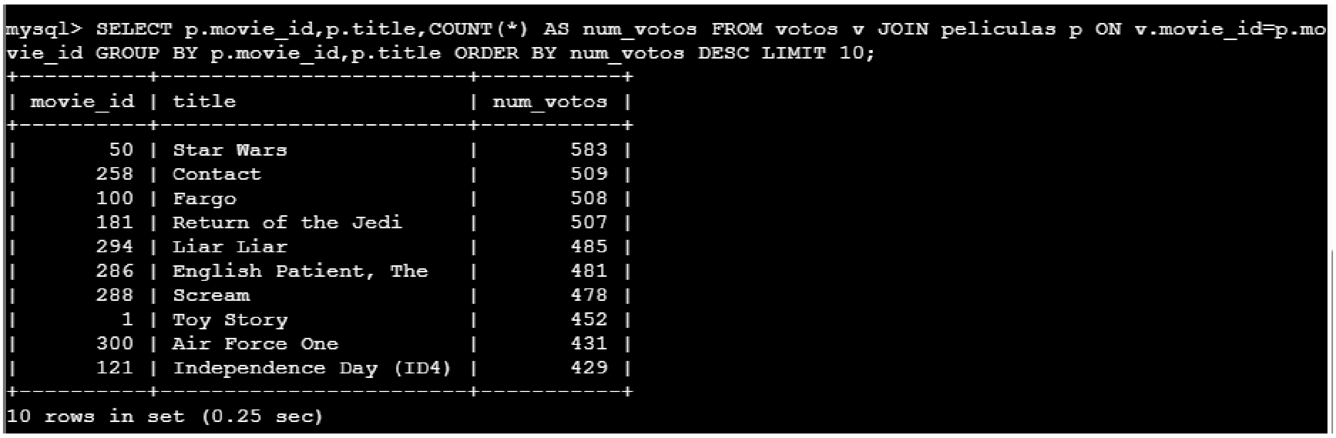
APARTADO D

INTRODUCCIÓN

- **Realiza las consultas de esta práctica utilizando DBeaver. Muestra la consulta y una captura donde se vea la salida de las mismas, al menos parcialmente.**
- **Ten en cuenta que la fecha de la votación está en formato `TIMESTAMP`.**

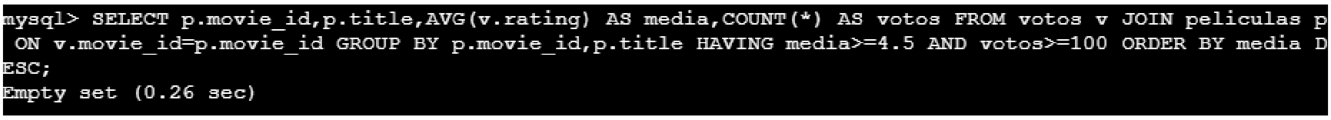
CONTENIDO

1. **Top 10 películas más votadas de todos los tiempos (número de votos, no el valor de este).**



```
mysql> SELECT p.movie_id,p.title,COUNT(*) AS num_votos FROM votos v JOIN peliculas p ON v.movie_id=p.mo
vie_id GROUP BY p.movie_id,p.title ORDER BY num_votos DESC LIMIT 10;
+----------+------------------------+-----------+
| movie_id | title                  | num_votos |
+----------+------------------------+-----------+
|       50 | Star Wars              |       583 |
|      258 | Contact                |       509 |
|      100 | Fargo                  |       508 |
|      181 | Return of the Jedi     |       507 |
|      294 | Liar Liar              |       485 |
|      286 | English Patient, The   |       481 |
|      288 | Scream                 |       478 |
|        1 | Toy Story              |       452 |
|      300 | Air Force One          |       431 |
|      121 | Independence Day (ID4) |       429 |
+----------+------------------------+-----------+
10 rows in set (0.25 sec)
```

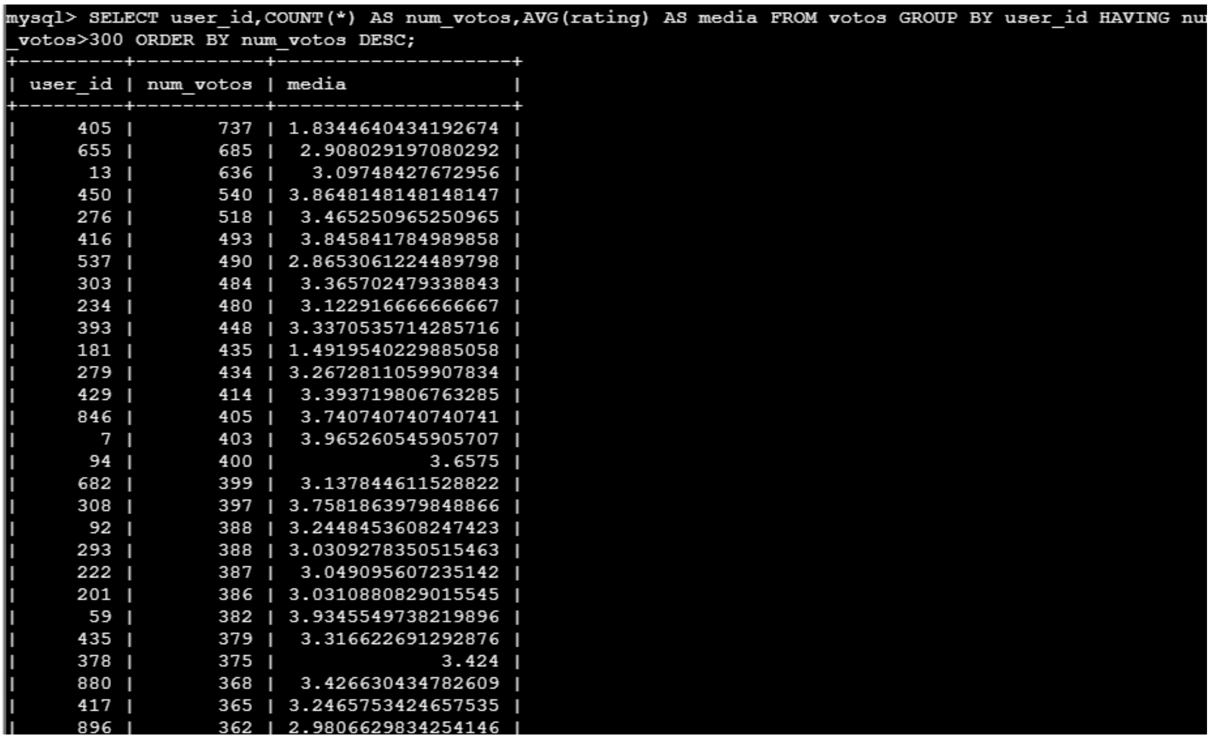
2. **Películas con nota media ≥ 4.5 y al menos 100 valoraciones.**



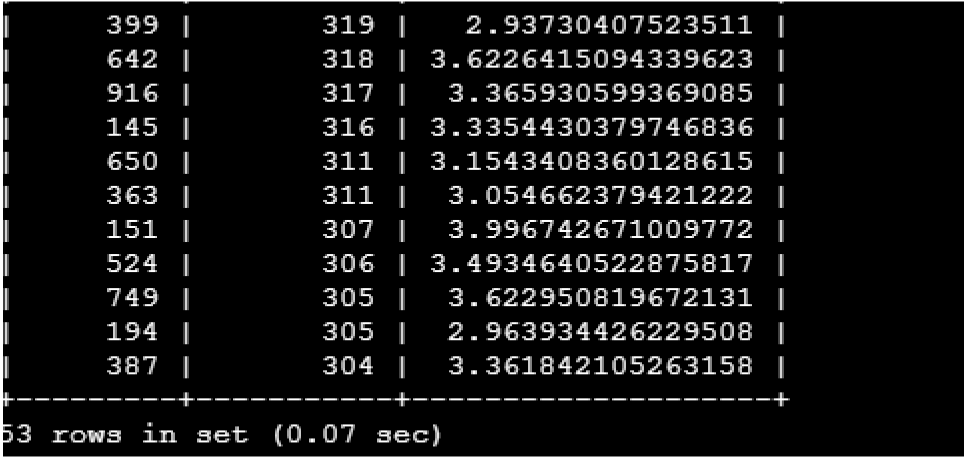
```
mysql> SELECT p.movie_id,p.title,AVG(v.rating) AS media,COUNT(*) AS votos FROM votos v JOIN peliculas p
 ON v.movie_id=p.movie_id GROUP BY p.movie_id,p.title HAVING media>=4.5 AND votos>=100 ORDER BY media D
ESC;
Empty set (0.26 sec)
```

No nos entrega ningún valor

3. **Usuarios que han dado más de 300 valoraciones y su nota media.**

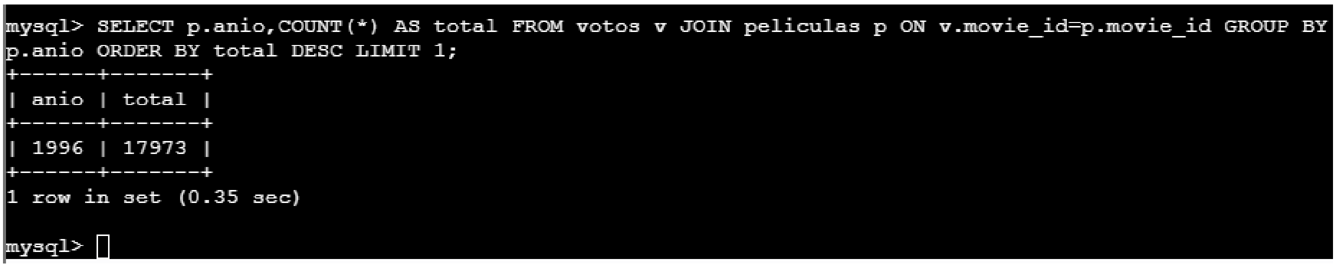


```
mysql> SELECT user_id,COUNT(*) AS num_votos,AVG(rating) AS media FROM votos GROUP BY user_id HAVING nu
_votos>300 ORDER BY num_votos DESC;
+---------+-----------+--------------------+
| user_id | num_votos | media              |
+---------+-----------+--------------------+
|     405 |       737 | 1.8344640434192674 |
|     655 |       685 |  2.908029197080292 |
|      13 |       636 |   3.09748427672956 |
|     450 |       540 | 3.8648148148148147 |
|     276 |       518 |  3.465250965250965 |
|     416 |       493 |  3.845841784989858 |
|     537 |       490 | 2.8653061224489798 |
|     303 |       484 |  3.365702479338843 |
|     234 |       480 |  3.122916666666667 |
|     393 |       448 | 3.3370535714285716 |
|     181 |       435 | 1.4919540229885058 |
|     279 |       434 | 3.2672811059907834 |
|     429 |       414 |  3.393719806763285 |
|     846 |       405 |  3.740740740740741 |
|       7 |       403 |  3.965260545905707 |
|      94 |       400 |             3.6575 |
|     682 |       399 |  3.137844611528822 |
|     308 |       397 | 3.7581863979848866 |
|      92 |       388 | 3.2448453608247423 |
|     293 |       388 | 3.0309278350515463 |
|     222 |       387 |  3.049095607235142 |
|     201 |       386 | 3.0310880829015545 |
|      59 |       382 | 3.9345549738219896 |
|     435 |       379 |  3.316622691292876 |
|     378 |       375 |              3.424 |
|     880 |       368 |  3.426630434782609 |
|     417 |       365 | 3.2465753424657535 |
|     896 |       362 | 2.9806629834254146 |
```



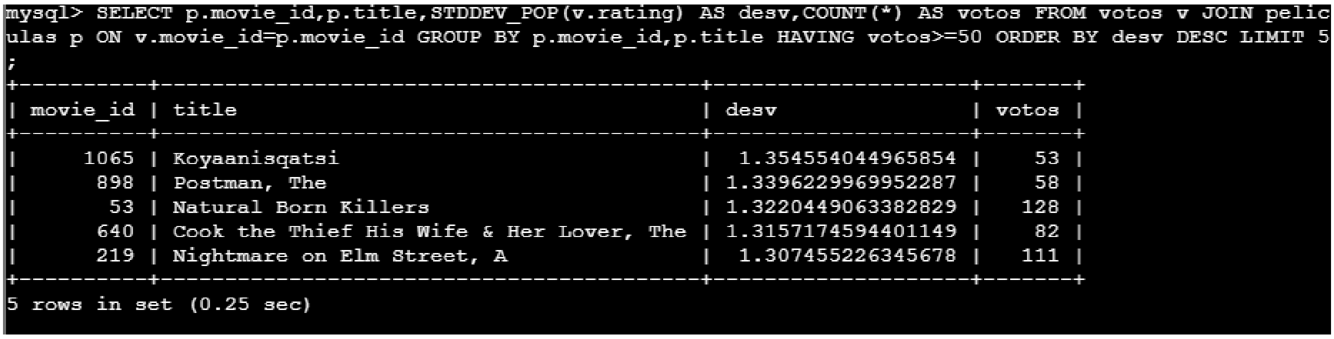
```
|     399 |       319 |   2.93730407523511 |
|     642 |       318 | 3.6226415094339623 |
|     916 |       317 |  3.365930599369085 |
|     145 |       316 | 3.3354430379746836 |
|     650 |       311 | 3.1543408360128615 |
|     363 |       311 |  3.054662379421222 |
|     151 |       307 |  3.996742671009772 |
|     524 |       306 | 3.4934640522875817 |
|     749 |       305 |  3.622950819672131 |
|     194 |       305 |  2.963934426229508 |
|     387 |       304 |  3.361842105263158 |
+---------+-----------+--------------------+
53 rows in set (0.07 sec)
```

4. **Año con más películas votadas (por número total de votos).**



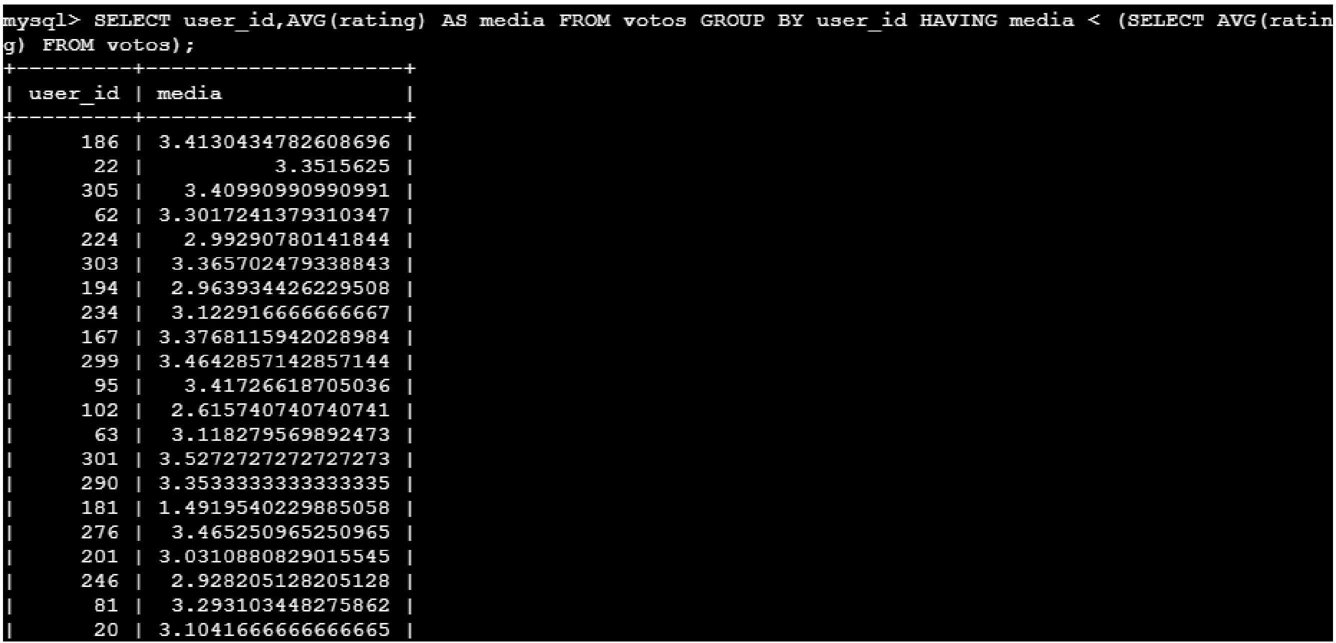
```
mysql> SELECT p.anio,COUNT(*) AS total FROM votos v JOIN peliculas p ON v.movie_id=p.movie_id GROUP BY
p.anio ORDER BY total DESC LIMIT 1;
+------+-------+
| anio | total |
+------+-------+
| 1996 | 17973 |
+------+-------+
1 row in set (0.35 sec)

mysql> 
```

5. **Las 5 películas más "*polarizadas*" (mayor desviación estándar, con valoraciones muy extremas) con al menos 50 votos. (Investiga qué función de SQL de da la desviación estándar).**

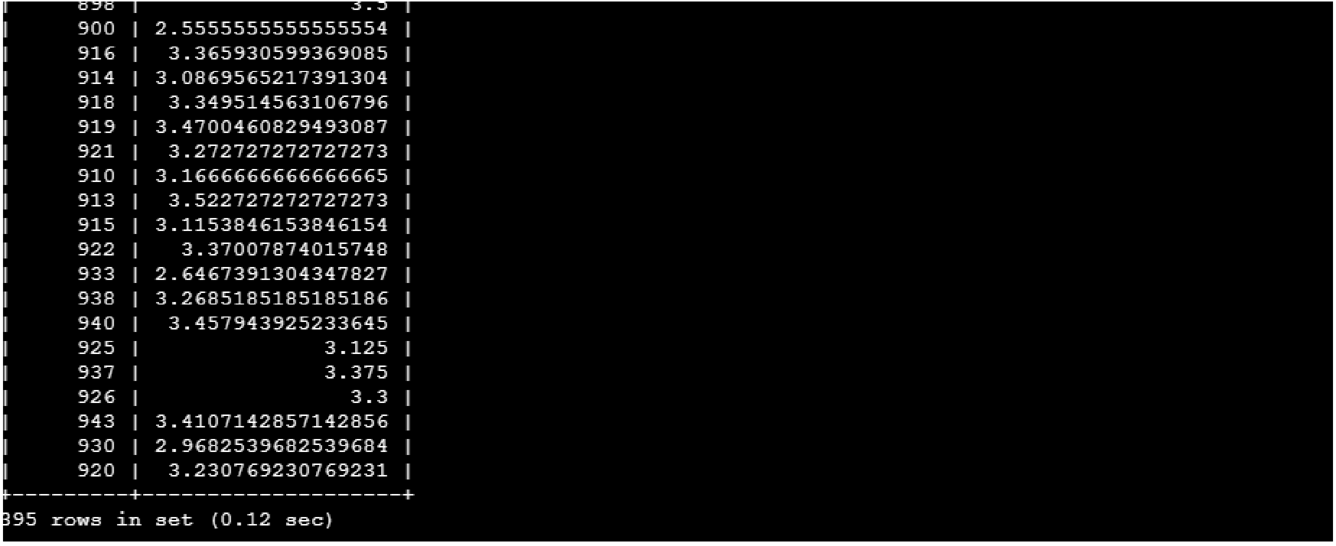


```
mysql> SELECT p.movie_id,p.title,STDDEV_POP(v.rating) AS desv,COUNT(*) AS votos FROM votos v JOIN pelic
ulas p ON v.movie_id=p.movie_id GROUP BY p.movie_id,p.title HAVING votos>=50 ORDER BY desv DESC LIMIT 5
;
+----------+------------------------------------------+--------------------+-------+
| movie_id | title                                    | desv               | votos |
+----------+------------------------------------------+--------------------+-------+
|     1065 | Koyaanisqatsi                            |  1.354554044965854 |    53 |
|      898 | Postman, The                             | 1.3396229969952287 |    58 |
|       53 | Natural Born Killers                     | 1.3220449063382829 |   128 |
|      640 | Cook the Thief His Wife & Her Lover, The | 1.3157174594401149 |    82 |
|      219 | Nightmare on Elm Street, A               |  1.307455226345678 |   111 |
+----------+------------------------------------------+--------------------+-------+
5 rows in set (0.25 sec)
```

6. **Usuarios cuya nota media es menor que la nota media global.**

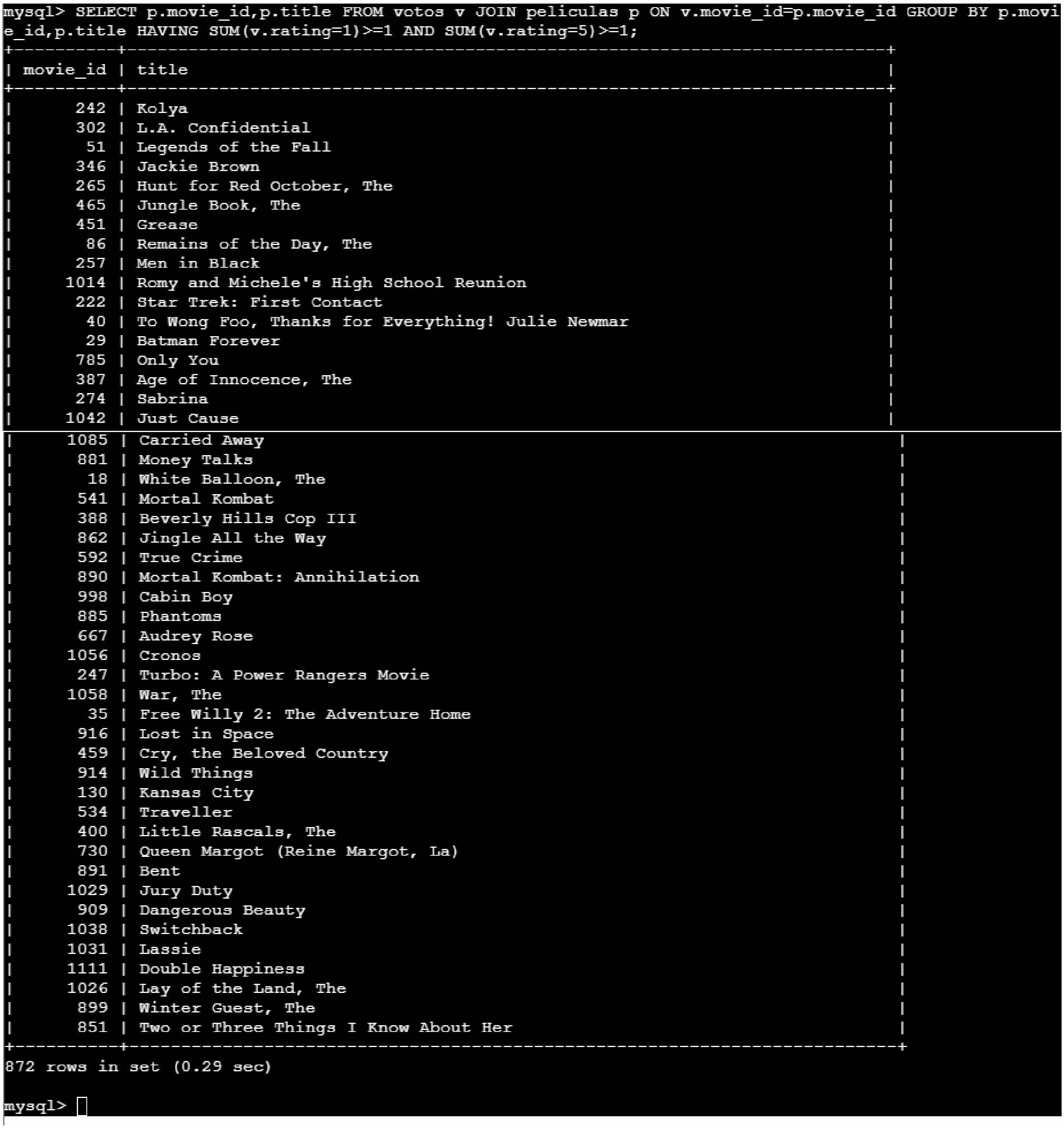


```
mysql> SELECT user_id,AVG(rating) AS media FROM votos GROUP BY user_id HAVING media < (SELECT AVG(ratin
g) FROM votos);
+---------+--------------------+
| user_id | media              |
+---------+--------------------+
|     186 | 3.4130434782608696 |
|      22 |          3.3515625 |
|     305 |   3.40990990990991 |
|      62 | 3.3017241379310347 |
|     224 |   2.99290780141844 |
|     303 |  3.365702479338843 |
|     194 |  2.963934426229508 |
|     234 |  3.122916666666667 |
|     167 | 3.3768115942028984 |
|     299 | 3.4642857142857144 |
|      95 |   3.41726618705036 |
|     102 |  2.615740740740741 |
|      63 |  3.118279569892473 |
|     301 | 3.5272727272727273 |
|     290 | 3.3533333333333335 |
|     181 | 1.4919540229885058 |
|     276 |  3.465250965250965 |
|     201 | 3.0310880829015545 |
|     246 |  2.928205128205128 |
|      81 |  3.293103448275862 |
|      20 | 3.1041666666666665 |
```



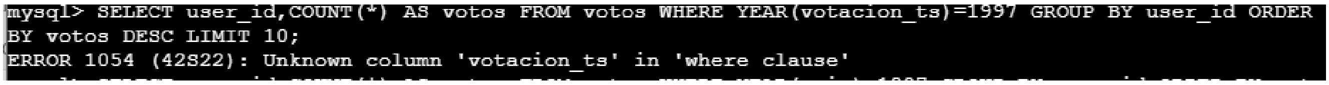
```
|     898 |                3.5 |
|     900 | 2.5555555555555554 |
|     916 |  3.365930599369085 |
|     914 | 3.0869565217391304 |
|     918 |  3.349514563106796 |
|     919 | 3.4700460829493087 |
|     921 |  3.272727272727273 |
|     910 | 3.1666666666666665 |
|     913 |  3.522727272727273 |
|     915 | 3.1153846153846154 |
|     922 |   3.37007874015748 |
|     933 | 2.6467391304347827 |
|     938 | 3.2685185185185186 |
|     940 |  3.457943925233645 |
|     925 |              3.125 |
|     937 |              3.375 |
|     926 |                3.3 |
|     943 | 3.4107142857142856 |
|     930 | 2.9682539682539684 |
|     920 |  3.230769230769231 |
+---------+--------------------+
395 rows in set (0.12 sec)
```

7. **Películas que han recibido al menos una valoración de 1 y una de 5 (las más divididas).**



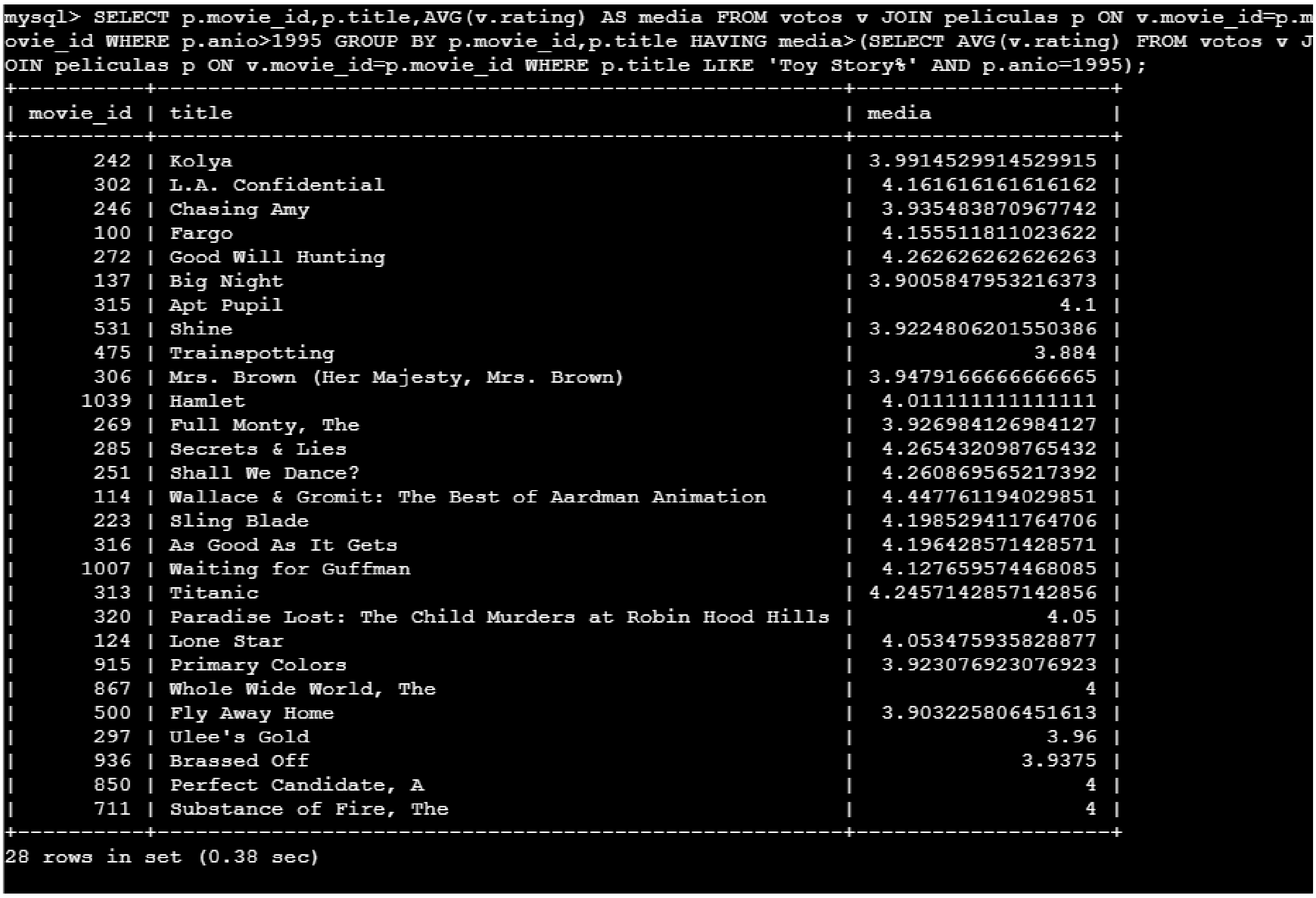
```
mysql> SELECT p.movie_id,p.title FROM votos v JOIN peliculas p ON v.movie_id=p.movie_id GROUP BY p.movi
e_id,p.title HAVING SUM(v.rating=1)>=1 AND SUM(v.rating=5)>=1;
+----------+--------------------------------------------------------------------------+
| movie_id | title                                                                    |
+----------+--------------------------------------------------------------------------+
|      242 | Kolya                                                                    |
|      302 | L.A. Confidential                                                        |
|       51 | Legends of the Fall                                                      |
|      346 | Jackie Brown                                                             |
|      265 | Hunt for Red October, The                                                |
|      465 | Jungle Book, The                                                         |
|      451 | Grease                                                                   |
|       86 | Remains of the Day, The                                                  |
|      257 | Men in Black                                                             |
|     1014 | Romy and Michele's High School Reunion                                   |
|      222 | Star Trek: First Contact                                                 |
|       40 | To Wong Foo, Thanks for Everything! Julie Newmar                         |
|       29 | Batman Forever                                                           |
|      785 | Only You                                                                 |
|      387 | Age of Innocence, The                                                    |
|      274 | Sabrina                                                                  |
|     1042 | Just Cause                                                               |
|     1085 | Carried Away                                                             |
|      881 | Money Talks                                                              |
|       18 | White Balloon, The                                                       |
|      541 | Mortal Kombat                                                            |
|      388 | Beverly Hills Cop III                                                    |
|      862 | Jingle All the Way                                                       |
|      592 | True Crime                                                               |
|      890 | Mortal Kombat: Annihilation                                              |
|      998 | Cabin Boy                                                                |
|      885 | Phantoms                                                                 |
|      667 | Audrey Rose                                                              |
|     1056 | Cronos                                                                   |
|      247 | Turbo: A Power Rangers Movie                                             |
|     1058 | War, The                                                                 |
|       35 | Free Willy 2: The Adventure Home                                         |
|      916 | Lost in Space                                                            |
|      459 | Cry, the Beloved Country                                                 |
|      914 | Wild Things                                                              |
|      130 | Kansas City                                                              |
|      534 | Traveller                                                                |
|      400 | Little Rascals, The                                                      |
|      730 | Queen Margot (Reine Margot, La)                                          |
|      891 | Bent                                                                     |
|     1029 | Jury Duty                                                                |
|      909 | Dangerous Beauty                                                         |
|     1038 | Switchback                                                               |
|     1031 | Lassie                                                                   |
|     1111 | Double Happiness                                                         |
|     1026 | Lay of the Land, The                                                     |
|      899 | Winter Guest, The                                                        |
|      851 | Two or Three Things I Know About Her                                     |
+----------+--------------------------------------------------------------------------+
872 rows in set (0.29 sec)

mysql> 
```

8. **Top 10 usuarios más activos en 1997 (por número de valoraciones ese año).**



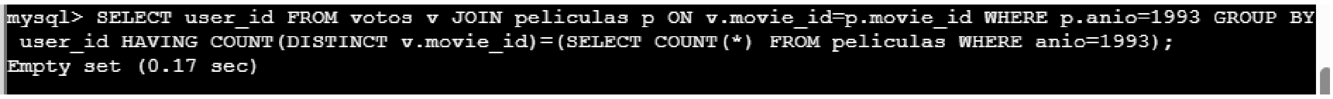
```
mysql> SELECT user_id,COUNT(*) AS votos FROM votos WHERE YEAR(votacion_ts)=1997 GROUP BY user_id ORDER
BY votos DESC LIMIT 10;
ERROR 1054 (42S22): Unknown column 'votacion_ts' in 'where clause'
```

9. **Películas estrenadas después de 1995 con mejor nota media que "Toy Story (1995).**



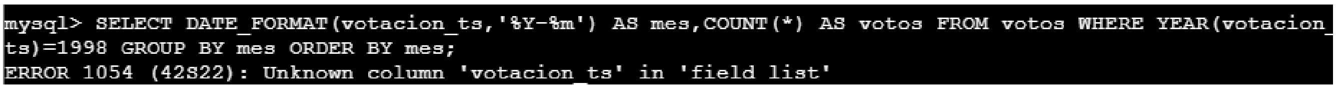
```
mysql> SELECT p.movie_id,p.title,AVG(v.rating) AS media FROM votos v JOIN peliculas p ON v.movie_id=p.m
ovie_id WHERE p.anio>1995 GROUP BY p.movie_id,p.title HAVING media>(SELECT AVG(v.rating) FROM votos v J
OIN peliculas p ON v.movie_id=p.movie_id WHERE p.title LIKE 'Toy Story%' AND p.anio=1995);
+----------+-------------------------------------------------------+--------------------+
| movie_id | title                                                 | media              |
+----------+-------------------------------------------------------+--------------------+
|      242 | Kolya                                                 | 3.9914529914529915 |
|      302 | L.A. Confidential                                     |  4.161616161616162 |
|      246 | Chasing Amy                                           |  3.935483870967742 |
|      100 | Fargo                                                 |  4.155511811023622 |
|      272 | Good Will Hunting                                     |  4.262626262626263 |
|      137 | Big Night                                             | 3.9005847953216373 |
|      315 | Apt Pupil                                             |                4.1 |
|      531 | Shine                                                 | 3.9224806201550386 |
|      475 | Trainspotting                                         |              3.884 |
|      306 | Mrs. Brown (Her Majesty, Mrs. Brown)                  | 3.9479166666666665 |
|     1039 | Hamlet                                                |  4.011111111111111 |
|      269 | Full Monty, The                                       |  3.926984126984127 |
|      285 | Secrets & Lies                                        |  4.265432098765432 |
|      251 | Shall We Dance?                                       |  4.260869565217392 |
|      114 | Wallace & Gromit: The Best of Aardman Animation       |  4.447761194029851 |
|      223 | Sling Blade                                           |  4.198529411764706 |
|      316 | As Good As It Gets                                    |  4.196428571428571 |
|     1007 | Waiting for Guffman                                   |  4.127659574468085 |
|      313 | Titanic                                               | 4.2457142857142856 |
|      320 | Paradise Lost: The Child Murders at Robin Hood Hills |               4.05 |
|      124 | Lone Star                                             |  4.053475935828877 |
|      915 | Primary Colors                                        |  3.923076923076923 |
|      867 | Whole Wide World, The                                 |                  4 |
|      500 | Fly Away Home                                         |  3.903225806451613 |
|      297 | Ulee's Gold                                           |               3.96 |
|      936 | Brassed Off                                           |             3.9375 |
|      850 | Perfect Candidate, A                                  |                  4 |
|      711 | Substance of Fire, The                                |                  4 |
+----------+-------------------------------------------------------+--------------------+
28 rows in set (0.38 sec)
```

10. **Usuarios que han valorado todas las películas estrenadas en 1993.**



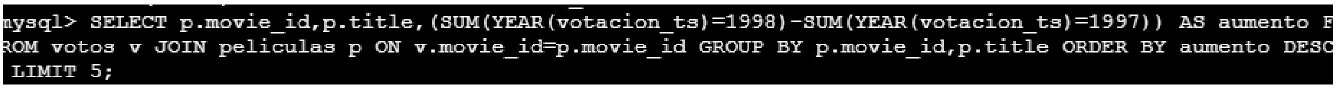
```
mysql> SELECT user_id FROM votos v JOIN peliculas p ON v.movie_id=p.movie_id WHERE p.anio=1993 GROUP BY
 user_id HAVING COUNT(DISTINCT v.movie_id)=(SELECT COUNT(*) FROM peliculas WHERE anio=1993);
Empty set (0.17 sec)
```

11. **Evolución mensual del número de valoraciones en 1998.**



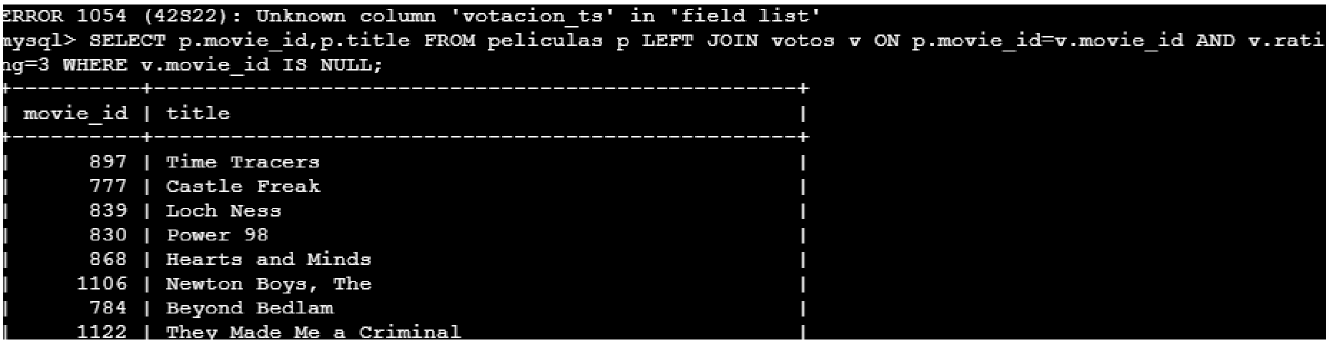
```
mysql> SELECT DATE_FORMAT(votacion_ts,'%Y-%m') AS mes,COUNT(*) AS votos FROM votos WHERE YEAR(votacion_
ts)=1998 GROUP BY mes ORDER BY mes;
ERROR 1054 (42S22): Unknown column 'votacion_ts' in 'field list'
```

12. **Las 5 películas con mayor aumento de popularidad (comparar 1997 vs 1998).**



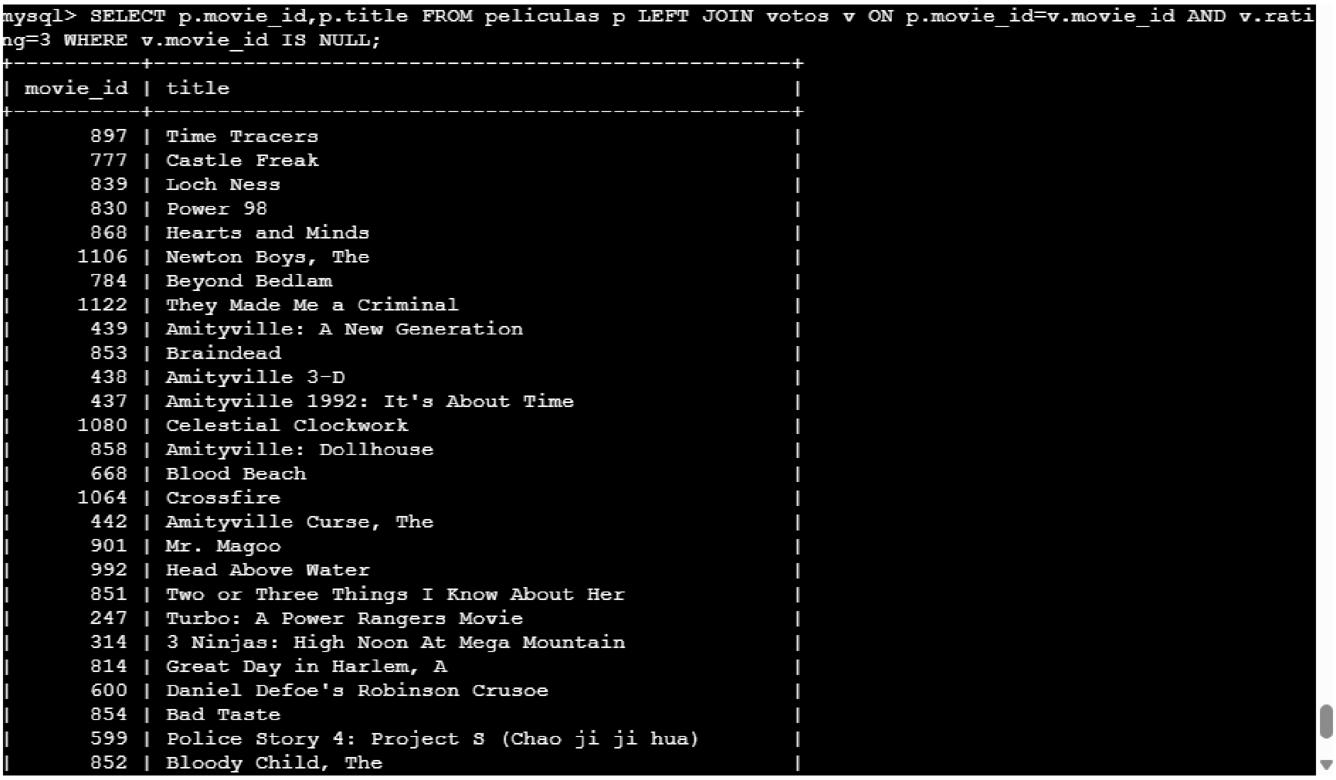
```
ysql> SELECT p.movie_id,p.title,(SUM(YEAR(votacion_ts)=1998)-SUM(YEAR(votacion_ts)=1997)) AS aumento F
ROM votos v JOIN peliculas p ON v.movie_id=p.movie_id GROUP BY p.movie_id,p.title ORDER BY aumento DESC
 LIMIT 5;
```

13. **Usuarios que han valorado más películas que la media de su género.**



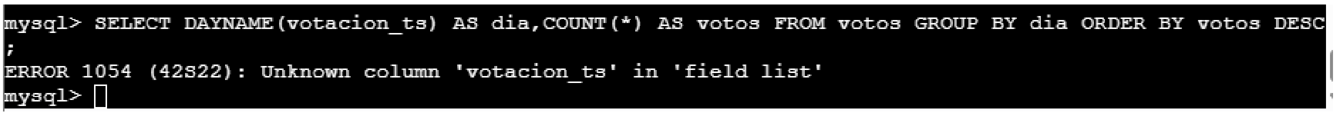
```
ERROR 1054 (42S22): Unknown column 'votacion_ts' in 'field list'
mysql> SELECT p.movie_id,p.title FROM peliculas p LEFT JOIN votos v ON p.movie_id=v.movie_id AND v.rati
ng=3 WHERE v.movie_id IS NULL;
+----------+---------------------------------------------------+
| movie_id | title                                             |
+----------+---------------------------------------------------+
|      897 | Time Tracers                                      |
|      777 | Castle Freak                                      |
|      839 | Loch Ness                                         |
|      830 | Power 98                                          |
|      868 | Hearts and Minds                                  |
|     1106 | Newton Boys, The                                  |
|      784 | Beyond Bedlam                                     |
|     1122 | They Made Me a Criminal                           |
```

14. **Películas que nadie ha valorado con 3 estrellas (solo 1,2,4,5).**



```
mysql> SELECT p.movie_id,p.title FROM peliculas p LEFT JOIN votos v ON p.movie_id=v.movie_id AND v.rati
ng=3 WHERE v.movie_id IS NULL;
+----------+---------------------------------------------------+
| movie_id | title                                             |
+----------+---------------------------------------------------+
|      897 | Time Tracers                                      |
|      777 | Castle Freak                                      |
|      839 | Loch Ness                                         |
|      830 | Power 98                                          |
|      868 | Hearts and Minds                                  |
|     1106 | Newton Boys, The                                  |
|      784 | Beyond Bedlam                                     |
|     1122 | They Made Me a Criminal                           |
|      439 | Amityville: A New Generation                      |
|      853 | Braindead                                         |
|      438 | Amityville 3-D                                    |
|      437 | Amityville 1992: It's About Time                  |
|     1080 | Celestial Clockwork                               |
|      858 | Amityville: Dollhouse                             |
|      668 | Blood Beach                                       |
|     1064 | Crossfire                                         |
|      442 | Amityville Curse, The                             |
|      901 | Mr. Magoo                                         |
|      992 | Head Above Water                                  |
|      851 | Two or Three Things I Know About Her              |
|      247 | Turbo: A Power Rangers Movie                      |
|      314 | 3 Ninjas: High Noon At Mega Mountain              |
|      814 | Great Day in Harlem, A                            |
|      600 | Daniel Defoe's Robinson Crusoe                    |
|      854 | Bad Taste                                         |
|      599 | Police Story 4: Project S (Chao ji ji hua)        |
|      852 | Bloody Child, The                                 |
```

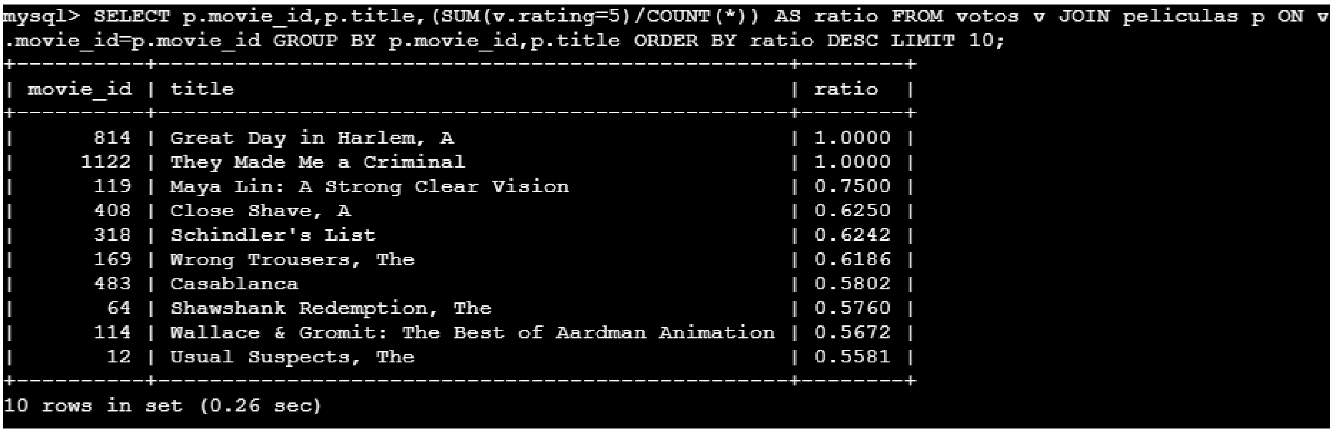
15. **Ranking de días de la semana con más actividad (lunes, martes...).**



```
mysql> SELECT DAYNAME(votacion_ts) AS dia,COUNT(*) AS votos FROM votos GROUP BY dia ORDER BY votos DESC
;
ERROR 1054 (42S22): Unknown column 'votacion_ts' in 'field list'
mysql> 
```

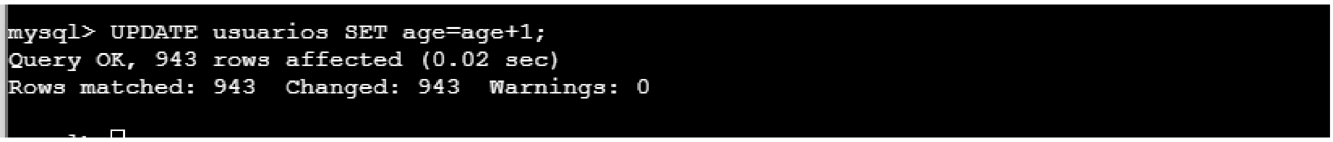
16. **Usuarios que han dado su primera y última valoración con diferencia > 6 meses.**

17. **Las 10 películas con mayor ratio 5-estrellas / total valoraciones.**



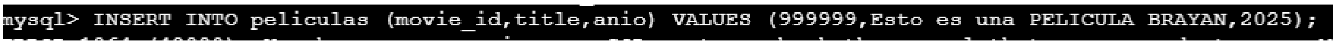
```
mysql> SELECT p.movie_id,p.title,(SUM(v.rating=5)/COUNT(*)) AS ratio FROM votos v JOIN peliculas p ON v
.movie_id=p.movie_id GROUP BY p.movie_id,p.title ORDER BY ratio DESC LIMIT 10;
+----------+-----------------------------------------------+--------+
| movie_id | title                                         | ratio  |
+----------+-----------------------------------------------+--------+
|      814 | Great Day in Harlem, A                        | 1.0000 |
|     1122 | They Made Me a Criminal                       | 1.0000 |
|      119 | Maya Lin: A Strong Clear Vision               | 0.7500 |
|      408 | Close Shave, A                                | 0.6250 |
|      318 | Schindler's List                              | 0.6242 |
|      169 | Wrong Trousers, The                           | 0.6186 |
|      483 | Casablanca                                    | 0.5802 |
|       64 | Shawshank Redemption, The                     | 0.5760 |
|      114 | Wallace & Gromit: The Best of Aardman Animation | 0.5672 |
|       12 | Usual Suspects, The                           | 0.5581 |
+----------+-----------------------------------------------+--------+
10 rows in set (0.26 sec)
```

18. **UPDATE: Aumenta en 1 año la edad de todos los usuarios (simulación de paso del tiempo).**



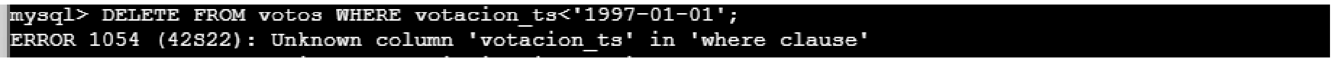
```
mysql> UPDATE usuarios SET age=age+1;
Query OK, 943 rows affected (0.02 sec)
Rows matched: 943  Changed: 943  Warnings: 0
```

19. **INSERT: Añade una nueva película ficticia estrenada hoy.**



```
mysql> INSERT INTO peliculas (movie_id,title,anio) VALUES (999999,Esto es una PELICULA BRAYAN,2025);
```

20. **DELETE: Elimina todas las valoraciones anteriores a 1997.**



```
mysql> DELETE FROM votos WHERE votacion_ts<'1997-01-01';
ERROR 1054 (42S22): Unknown column 'votacion_ts' in 'where clause'
```