



PROYECTO FINAL BASE DE DATOS

DAVID ERNESTO CARRION
ACOSTA

1. Importar el archivo

<< < 4.800 rows > >>

+ -

Tx: Auto ▾

DDL 🔍

WHERE

ORDER BY

```
production_companies
```

```
[{"name": "Metakwon Filmworks", "id": 65815}]
```

genres

Action Adventure Fantasy Science Fiction

Adventure Fantasy Action

```
homepage
<null>
<null>
<null>
```

[illegible]

2. Normalizar Columnas con Campos repetitivos

	status
1	Post Production
2	Released
3	Rumored

	job
1	24 Frame Playback
2	2D Artist
3	2D Supervisor
4	3D Animator
5	3D Artist
6	3D Coordinator
7	3D Modeller
8	3D Supervisor

	department
1	Actors
2	Art
3	Camera
4	Costume & Make-Up
5	Crew
6	Directing
7	Editing
8	Lighting
9	Production
10	Sound
11	Visual Effects
12	Writing


	gender
1	0
2	1
3	2

```
10 DROP TABLE IF EXISTS status;
11 CREATE TABLE status (status VARCHAR(100)) AS SELECT DISTINCT status
12 FROM movie_dataset.movie_dataset;
13
14 ALTER TABLE status
15 ADD PRIMARY KEY (status);
16
17 -- Creación de la relación movie-status
18 CREATE TABLE status_movie (id INT NOT NULL,
19                               status VARCHAR(100)) AS SELECT id, st.status AS status
20 FROM movie_dataset.movie_dataset movie_status, status st
21 WHERE movie_status.status = st.status;
22
23 ALTER TABLE status_movie
24 ADD FOREIGN KEY (id) REFERENCES movie(id),
25 ADD FOREIGN KEY (status) REFERENCES status(status);
```

3. Columnas con campos no atómicos

 genres

Romance Comedy Music

 cast

Michael Madsen Iman Crosson Lazar Ristovski

```
29 CREATE TABLE genres(genres VARCHAR(100)) AS
30     SELECT DISTINCT (
31         SUBSTRING_INDEX(SUBSTRING_INDEX(genres, ' ', 5), ' ', -1)) AS genre
32 FROM movie_dataset.movie_dataset;
33
34 DELETE
35 FROM genres
36 WHERE genres IS NULL;
37
38 ALTER TABLE genres
39     ADD PRIMARY KEY (genres);
40
41 CREATE TABLE genres_movie AS
42     SELECT tg.genres, id
43     FROM genres tg, movie_dataset.movie_dataset mv
44     WHERE INSTR(mv.genres, tg.genres )>0;
45
46 ALTER TABLE genres_movie
47     ADD FOREIGN KEY (id)
48     REFERENCES movie(id),
49     ADD FOREIGN KEY (genres)
50     REFERENCES genres(genres);
```



```

136 UPDATE movie_dataset.movie_dataset
137 SET cast = REPLACE(cast, 'Christopher Robin Miller', 'Christopher Robin_Miller' ),
138     cast = REPLACE(cast, 'Anna Claire Sneed', 'Anna Claire_Sneed' ),
139     cast = REPLACE(cast, 'Edward James Olmos', 'Edward James_Olmos' ),
140     cast = REPLACE(cast, 'Motwani Santhanam VTV', 'Motwani Santhanam' ),
141     cast = REPLACE(REPLACE(cast, ' de ', ' de' ), ' De ', ' De'),
321     cast = REPLACE(cast, 'Scott Chris Klein Thomas', 'Scott Chris Klein Thomas Thomas'),
322     cast = REPLACE(cast, 'Biggs Alyson Hannigan', 'Biggs Alyson_Hannigan'),
323     cast = REPLACE(cast, 'Anh Le Guy Pearce', 'Anh Le_Guy Pearce'),
324     cast = REPLACE(cast, 'Haley Jonny Lee Miller', 'Haley Jonny Miller'),
325     cast = REPLACE(cast, 'Spears Donovan Womack', 'Spears Donovan Womack');

```

Output Result 1		
4,045 rows		
	id	cast
1	19995	Sam Worthington Zoe Saldana Sigourney Weaver St...
2	285	Johnny Depp Orlando Bloom Keira Knightley Stell...
3	206647	Daniel Craig Christoph Waltz L\u00e9a Seydoux R...
4	49026	Christian Bale Michael Caine Gary Oldman Anne H...
5	49529	Taylor Kitsch Lynn Collins Samantha Morton Will...
6	559	Tobey Maguire Kirsten Dunst James Franco Thomas...
7	38757	Zachary Levi Mandy Moore Donna Murphy Ron Perlm...
8	99861	Robert DowneyJr. Chris Hemsworth Mark Ruffalo C...
9	767	Daniel Radcliffe Rupert Grint Emma Watson Tom F...
10	209112	Ben Affleck Henry Cavill Gal Gadot Amy Adams Je...
11	1452	Brandon Routh Kevin Spacey Kate Bosworth James ...
12	10764	Daniel Craig Olga Kurylenko Mathieu Amalric Jud...
13	58	Johnny Depp Orlando Bloom Keira Knightley Stell...
14	57201	Johnny Depp Armie Hammer William Fichtner Helen...
15	49521	Henry Cavill Amy Adams Michael Shannon Kevin Co...
16	2454	Ben Barnes William Moseley Anna Popplewell Skan...
17	24428	Robert DowneyJr. Chris Evans Mark Ruffalo Chris...
18	1865	Johnny Depp Pen\u00e9lope Cruz Ian McShane Kevi...

4. Columnas con gran cantidad de nulos

1.691 rows	
WHERE	
ORDER BY	
homepage	
1	http://2016themovie.com/
2	http://300themovie.warnerbros.com
3	http://30minutesorless.com/
4	http://500days.com
5	http://50shadesofblackmovie.com/
6	http://8x.com/onenight/main.html
7	http://a24films.com/films/witch/
8	http://againstthewild.com/
9	http://albertnobbbs-themovie.com
10	http://alexandemmamovie.warnerbros.com/
11	http://allhatthemovie.com/
12	http://alltheprettyhorses.com/
13	http://alphaandomega3d.com/index.html
14	http://amostviolentyear.com/
15	http://angeleyesmovie.warnerbros.com/index_noflash.html
16	http://annabellemovie.com/
17	http://apollo18movie.net/
18	http://argothemovie.warnerbros.com/
19	http://asoundofthunder.warnerbros.com/
20	http://attacktheblock.com/
21	http://augustrushmovie.warnerbros.com/
22	http://baahubali.com/
23	http://bandslam-movie.com/
24	http://barakasamsara.com/
25	http://becomingjane-themovie.com/
26	http://beyondthelightsmovie.tumblr.com/


```
52  -- ----- Normalización de Homepage -----
53  DROP TABLE IF EXISTS homepage;
54  CREATE TABLE homepage(homepage VARCHAR(200)) AS
55  SELECT DISTINCT homepage
56  FROM movie_dataset.movie_dataset;
57
58  DELETE
59  FROM homepage
60  WHERE homepage IS NULL;
61
62  ALTER TABLE homepage
63  ADD PRIMARY KEY (homepage);
64
65
66  CREATE TABLE relacion_homepage (id INT NOT NULL PRIMARY KEY, homepage VARCHAR(200)) AS
67  SELECT DISTINCT id, homepage
68  FROM movie_dataset.movie_dataset
69  WHERE homepage != '' OR homepage IS NOT NULL;
70  -- Agregamos las claves foráneas de la relacion_homepage a movie y a homepage según corresponda.
71  ALTER TABLE relacion_homepage
72  ADD FOREIGN KEY (id)
73  REFERENCES movie(id),
74  ADD FOREIGN KEY (homepage)
75  REFERENCES homepage(homepage);
76
```

5. Columna Con Formato JSON

spoken_languages

```
[{"iso_639_1": "en", "name": "English"}, {"iso_639_1": "..."}, {"iso_639_1": "en", "name": "English"}]
```

production_countries

```
[{"iso_3166_1": "US", "name": "United States of America..."}, {"iso_3166_1": "US", "name": "United States of America..."}, {"iso_3166_1": "GB", "name": "United Kingdom"}, {"iso_3166_1": "..."}]
```

production_companies

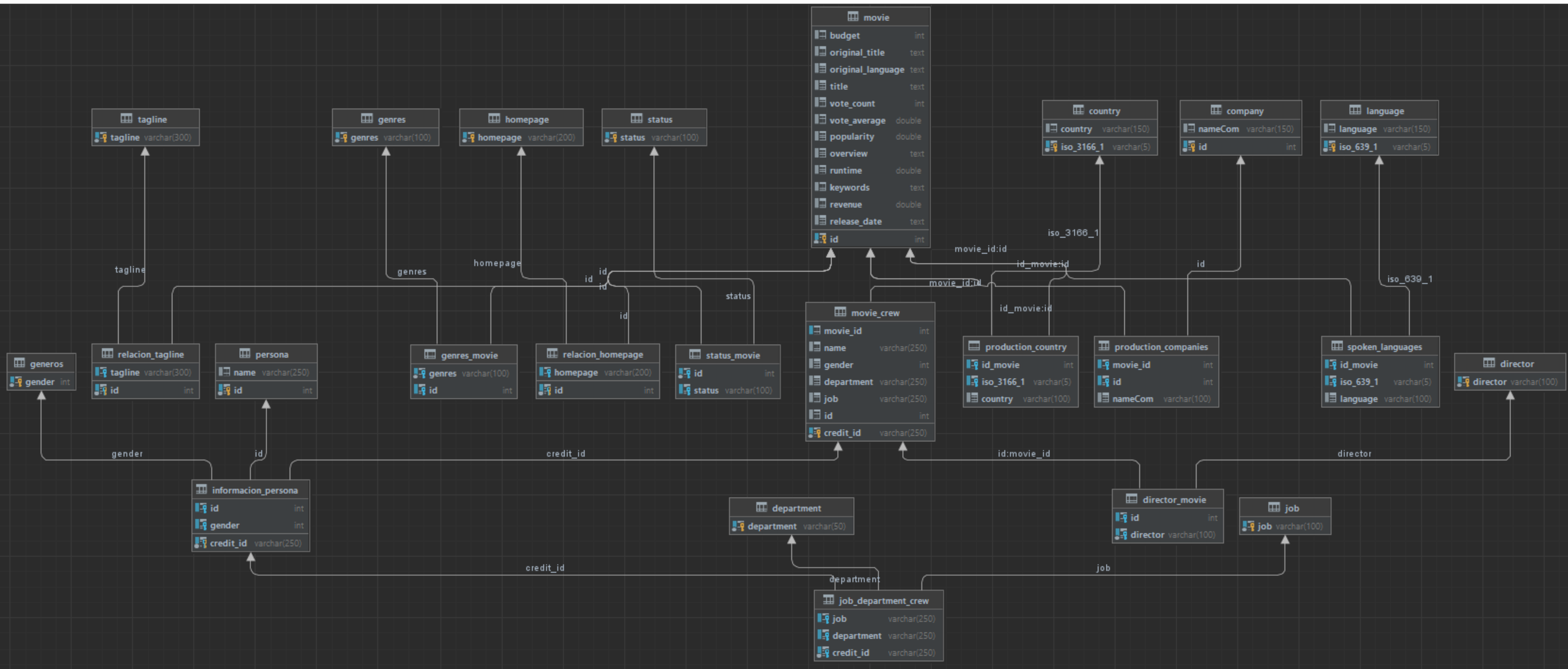
```
[{"name": "Ingenious Film Partners", "id": 289}, {"name": "..."}, {"name": "Walt Disney Pictures", "id": 2}, {"name": "J..."}, {"name": "Columbia Pictures", "id": 5}, {"name": "Danjaq'"}]
```

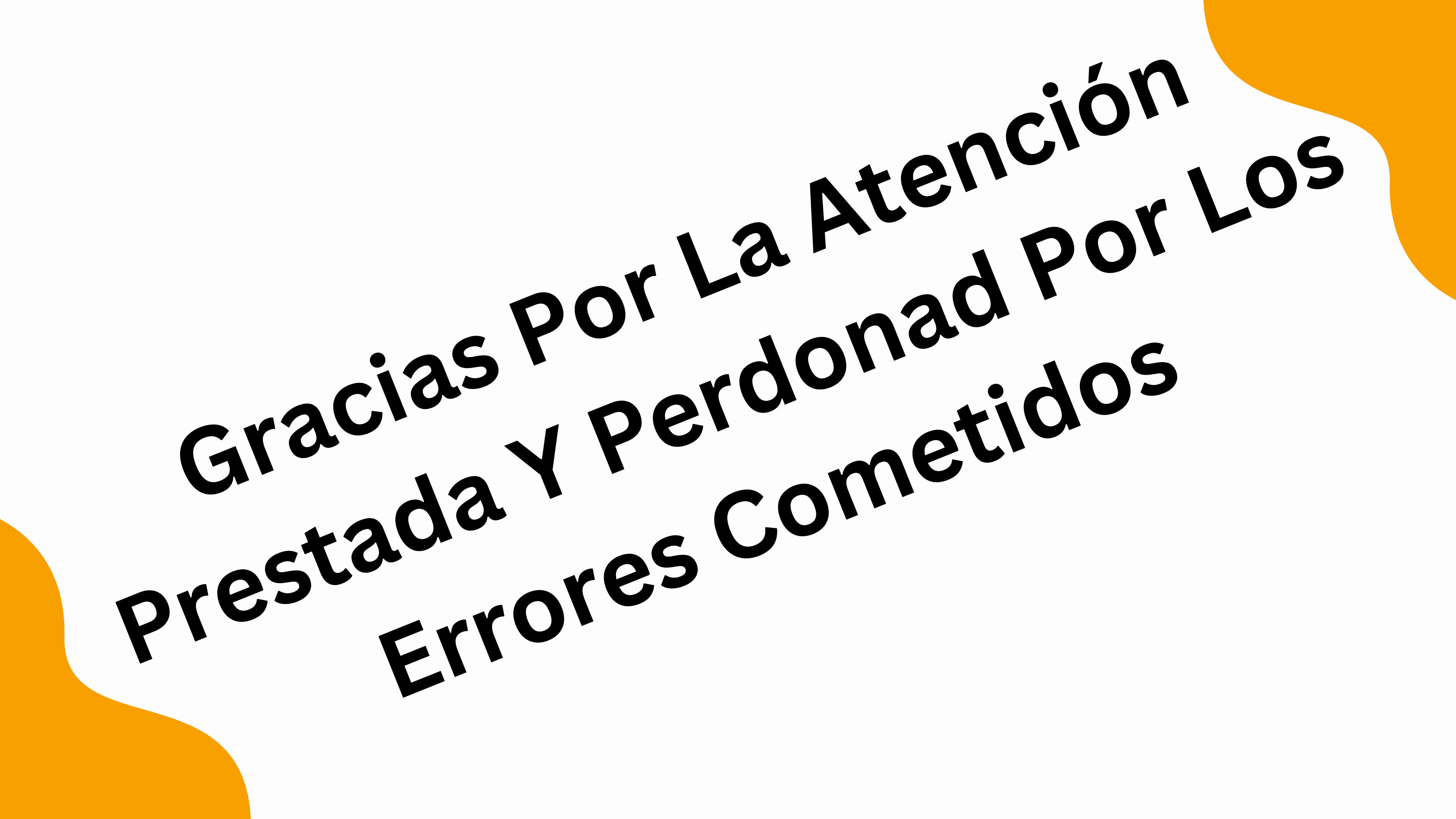
```
403      # [{"iso_639_1": "en", "name": "English"}]
404
405      DROP PROCEDURE IF EXISTS spoken_languages;
406      DELIMITER $$
407      CREATE PROCEDURE spoken_languages()
408      BEGIN
409          DECLARE done INT DEFAULT FALSE ;
410          DECLARE jsonData json ;
411          DECLARE jsonId varchar(250) ;
412          DECLARE jsonLabel varchar(250) ;
413          DECLARE movieId INT;
414          DECLARE i INT;
415
416          -- Declarar el cursor
417          DECLARE myCursor
418              CURSOR FOR
419              SELECT id, JSON_EXTRACT(CONVERT(spoken_languages USING UTF8MB4), '$[*]')
420              FROM movie_dataset.movie_dataset;
421          -- Declarar el handler para NOT FOUND
422
423          DECLARE CONTINUE HANDLER
424              FOR NOT FOUND SET done = TRUE ;
```

```
16      -- Declarar el cursor
17      DECLARE myCursor
18          CURSOR FOR
19          SELECT id, JSON_EXTRACT(CONVERT(spoken_languages USING UTF8MB4), '$[*]')
20          FROM movie_dataset.movie_dataset;
21      -- Declarar el handler para NOT FOUND
22
23      DECLARE CONTINUE HANDLER
24      FOR NOT FOUND SET done = TRUE ;
25
26      -- Abrir el cursor
27      OPEN myCursor ;
28      cursorLoop: LOOP
29
30          FETCH myCursor INTO movieId, jsonData;
31          SET i = 0;
32          IF done THEN
33              LEAVE cursorLoop ;
34          END IF ;
```

```
436 WHILE(JSON_EXTRACT(jsonData, CONCAT('$[', i, ']')) IS NOT NULL) DO
437     SET jsonId = IFNULL(JSON_EXTRACT(jsonData, CONCAT('$[', i, '].iso_639_1')), '') ;
438     SET jsonLabel = IFNULL(JSON_EXTRACT(jsonData, CONCAT('$[', i, '].name')), '') ;
439     SET @sql_text = CONCAT('INSERT INTO spoken_languages VALUES (', movieId, ', ', REPLACE(jsonId, '\\', '\\'), ', ', jsonLabel, '); ')
440
441     PREPARE stmt FROM @sql_text;
442     EXECUTE stmt;
443     DEALLOCATE PREPARE stmt;
444     SET i = i + 1;
445 END WHILE;
446
447 END LOOP ;
448
449 CLOSE myCursor ;
450
451 END$$
452 DELIMITER ;
453
454 DROP TABLE IF EXISTS spoken_languages;
455 CREATE TABLE spoken_languages (
456     id_movie INT,
457     iso_639_1 VARCHAR(5),
458     language VARCHAR(100));
459
460 CALL spoken_languages();
```

Resultado





**Gracias Por La Atención
Prestada Y Perdonad Por Los
Errores Cometidos**