FELIPE JIMENEZ

PRESENTACIÓN

PROYECTO FINAL

FUNDAMENTOS DE BASE DE DATOS

UNIVERSIDAD TÉCNICA
PARTICULAR DE LOJA

INTRODUCCIÓN

Este trabajo tiene como objetivo la creación de una base de datos relacional a partir de un archivo con datos crudos pertenecientes a **películas.** A continuación se presentan los pasos a seguir:

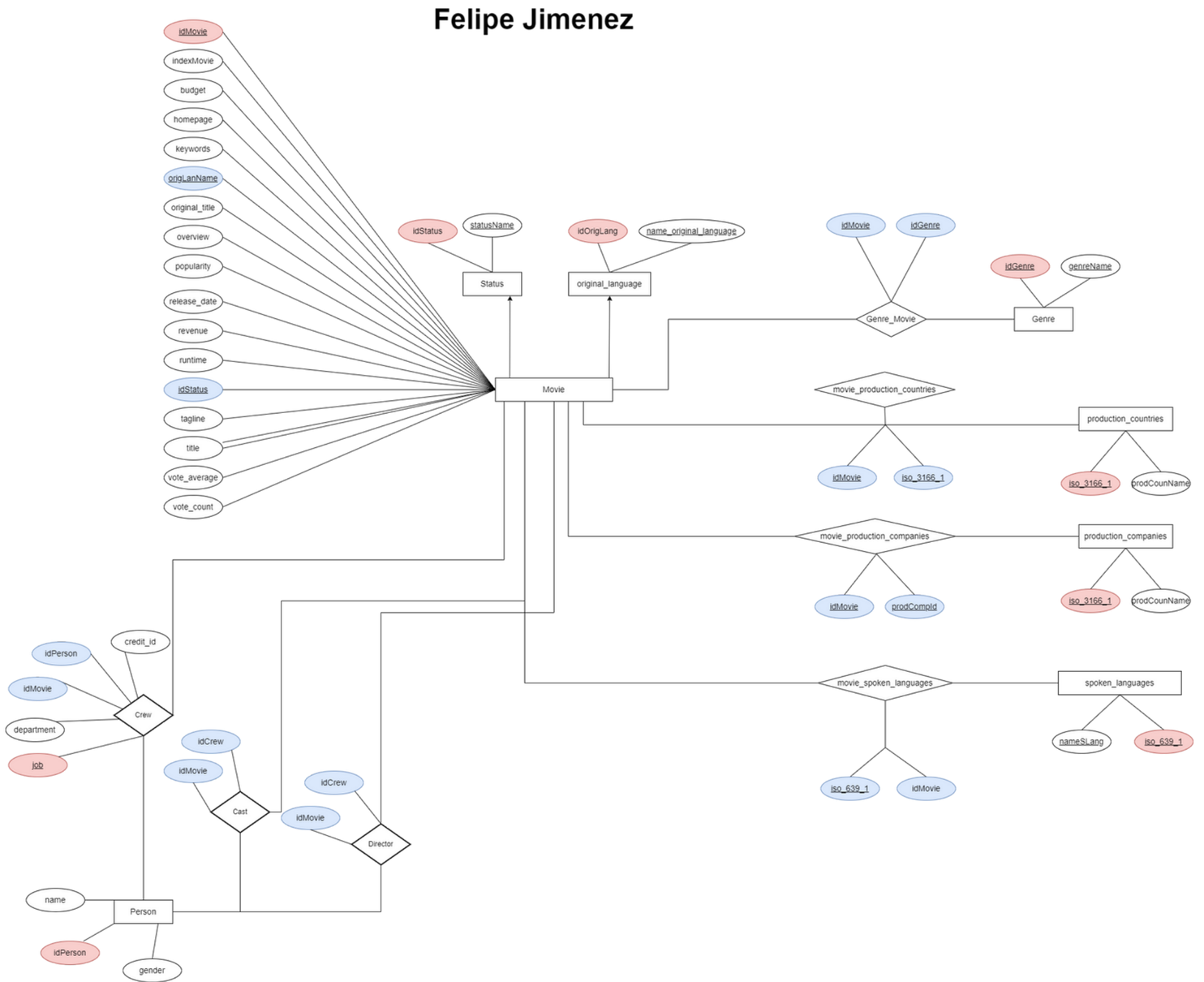
PASO 1. NORMALIZACIÓN DE LAS COLUMNAS

1 PRIMERA FORMA NORMAL

2 SEGUNDA FORMA NORMAL

3 TERCERA FORMA NORMAL

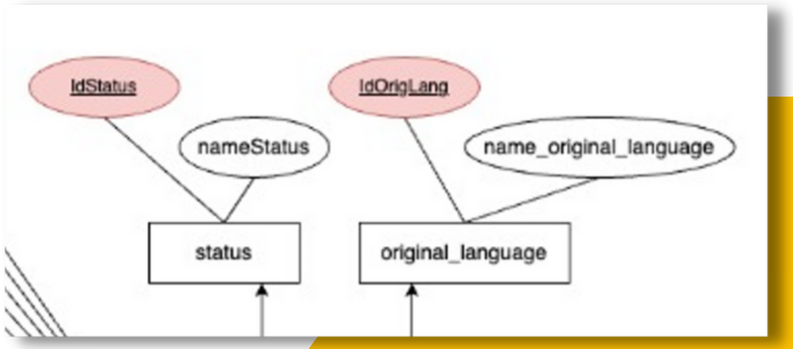
2. MODELO CONCEPTUAL



RELACIONES EXISTENTES

UNO A MUCHOS

- Crear una tabla para cada entidad o elemento en la relación.
- Asignar una clave primaria única a cada tabla.
- Crear una clave foránea en la tabla "muchos" que haga referencia a la clave primaria de la tabla "1".



MUCHOS A MUCHOS

- Crear una tabla para cada entidad o elemento en la relación.
- Crear una tercera tabla que actúe como tabla intermediaria entre las dos tablas originales.
- Asignar claves foráneas a esta tercera tabla que hagan referencia a las claves primarias de las tablas originales.

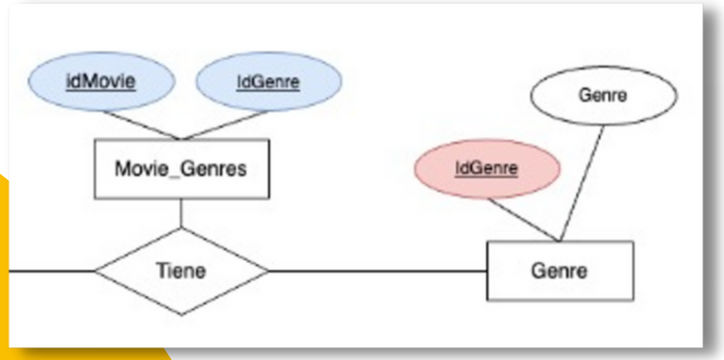
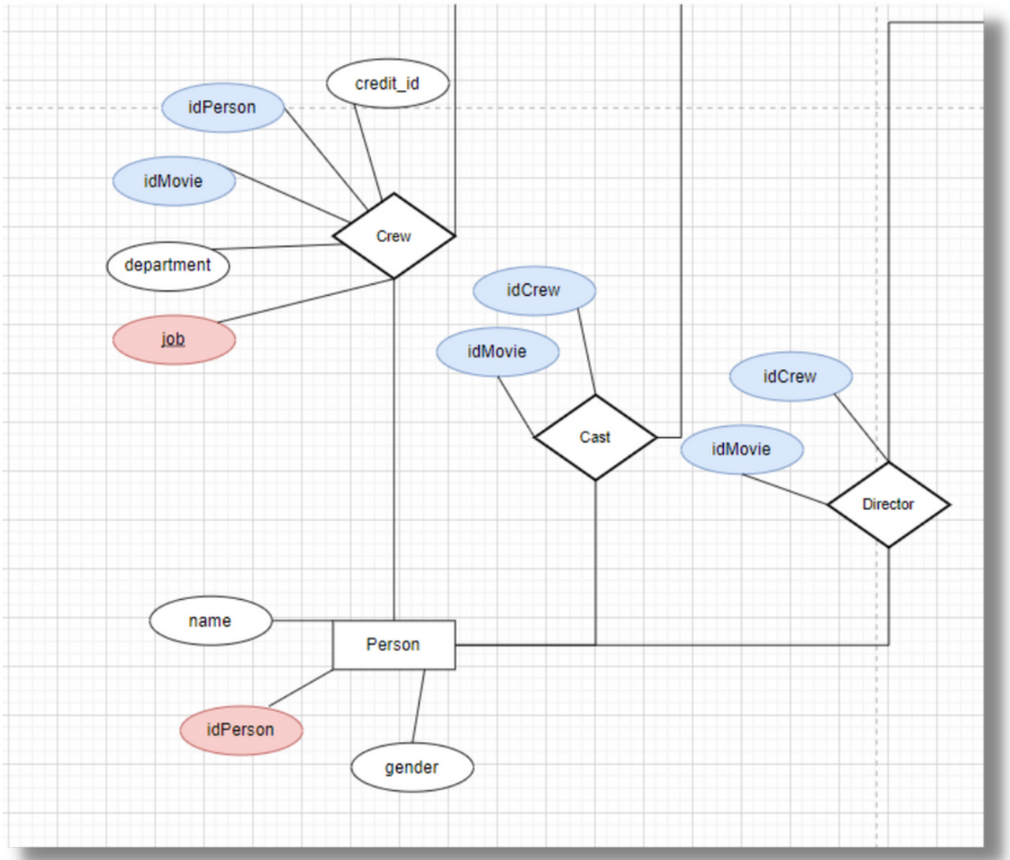
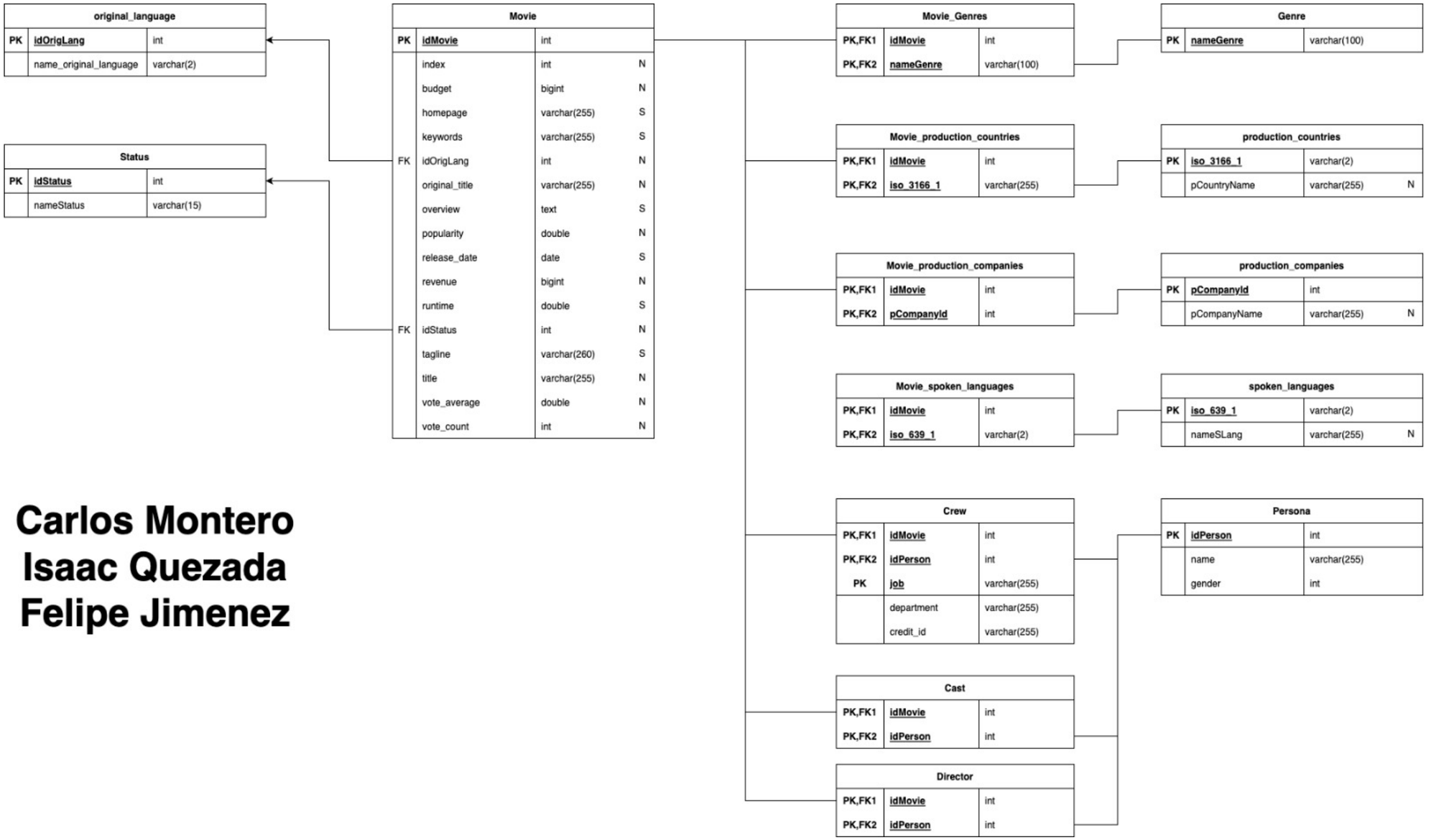


TABLA PERSONA

‘Cast’, ‘Crew’ y ‘Director’ contienen los nombres de integrantes que participan de alguna forma en la película. Estos necesitan ser normalizados en una tabla con datos comunes, en este caso llamada Persona.

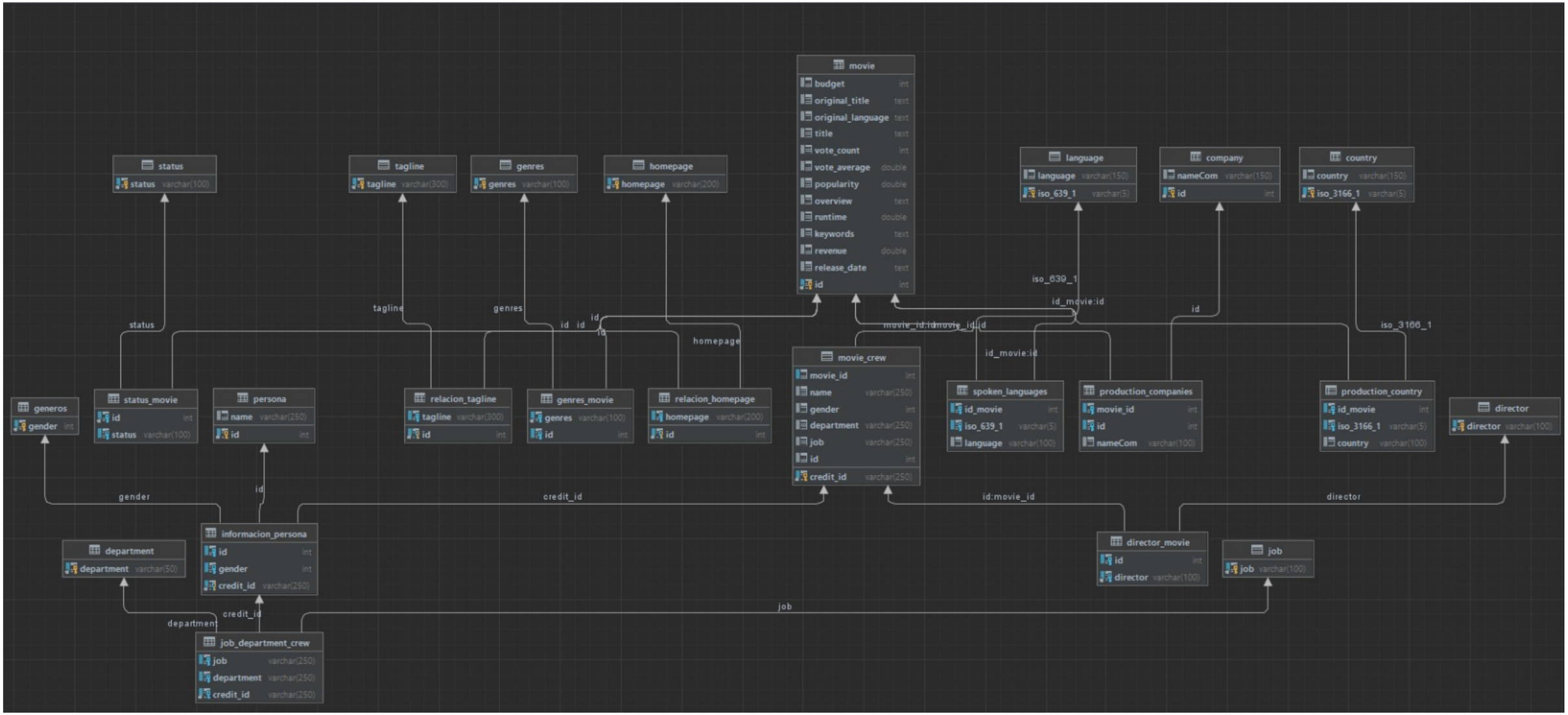


3. MODELO LÓGICO



Carlos Montero
Isaac Quezada
Felipe Jimenez

4. MODELO FÍSICO



6. CURSOR DE ORIGINAL_LANGUAGE

```
OPEN CursorOL;
CursorOL_loop: LOOP
    FETCH CursorOL INTO nameOL;

-- Si alcanzo el final del cursor entonces salir del ciclo repetitivo
    IF done THEN
        LEAVE CursorOL_loop;
    END IF;
    IF nameOL IS NULL THEN
        SET nameOL = '';
    END IF;
    SET @_oStatement = CONCAT('INSERT INTO original_languageCURSOR (name) VALUES (\'',
    nameOL,'\');');
    PREPARE sent1 FROM @_oStatement;
    EXECUTE sent1;
    DEALLOCATE PREPARE sent1;
```

```
DROP TABLE IF EXISTS original_languageCURSOR;

CREATE TABLE original_languageCURSOR (
    name varchar(255) PRIMARY KEY
);

SELECT * FROM original_languageCURSOR;
```

6. CURSOR DE PRODUCTION_COUNTRIES

```
cursorLoop: LOOP

    FETCH myCursor INTO idMovie, idProdCoun;

 -- Controlador para buscar cada uno de los arrays
  SET i = 0;

 -- Si alcanzo el final del cursor entonces salir del ciclo repetitivo
 IF done THEN
  LEAVE  cursorLoop ;
 END IF ;

 WHILE(JSON_EXTRACT(idProdCoun, CONCAT('$[', i, '].iso_3166_1')) IS NOT NULL) DO

 SET idJSON = JSON_EXTRACT(idProdCoun,  CONCAT('$[', i, '].iso_3166_1')) ;
 SET i = i + 1;

 SET @sql_text = CONCAT('INSERT INTO MovieProdCompTemp VALUES (', idMovie, ', ', REPLACE(idJSON,'\"',''), '); ');
   PREPARE stmt FROM @sql_text;
   EXECUTE stmt;
   DEALLOCATE PREPARE stmt;

 END WHILE;

END LOOP ;
```

7. MIGRACION DE DATOS

```
INSERT INTO `original_language`(`name_original_language`)
SELECT `name`
FROM original_languageCURSOR;
```

```
INSERT INTO `Status`(`nameStatus`)
SELECT `name`
FROM statusCURSOR;
```

```
INSERT INTO Movie (
                    idMovie, budget, homepage, original_title,
                    overview, popularity, release_date, revenue,
                    runtime, tagline, title, vote_average,
                    vote_count)

SELECT              id, budget, homepage, original_title, overview,
                    popularity, release_date, revenue, runtime,
                    tagline, title, vote_average, vote_count
FROM movie_dataset.movie_dataset_crudo;
```

GRACIAS POR SU ATENCIÓN