

**Programación de base de datos**

Avance 2.

**Estudiante:**

Carlos Andrés Cruz Calvo

Luis Fernando Solano Montoya

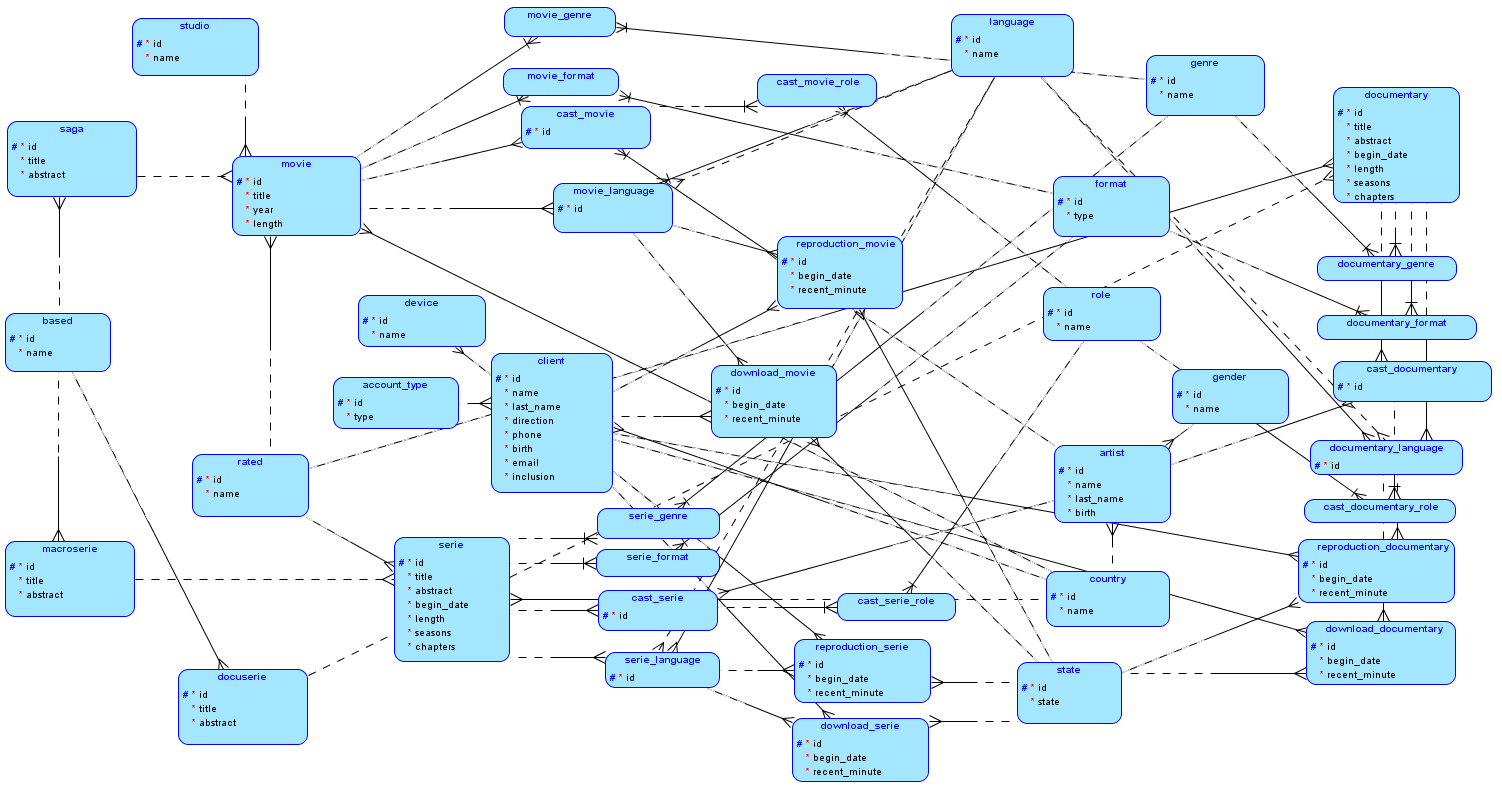
**Profesor:**

Ing. Luis Alberto Monge Fuentes

Universidad Cenfotec

lI Cuatrimestre – 2023

Diseño de base de datos: Modelo lógico



Modelo entidad-relación

A diagram of a computer

Description automatically generated

Script de la base de datos

-- Generated by Oracle SQL Developer Data Modeler 20.2.0.167.1538

--   at:        2023-07-04 18:00:03 CST

--   site:      Oracle Database 11g

--   type:      Oracle Database 11g

-- predefined type, no DDL - MDSYS.SDO\_GEOMETRY

-- predefined type, no DDL - XMLTYPE

CREATE TABLE account\_type (

    id    INTEGER NOT NULL,

    type  VARCHAR2(15) NOT NULL

);

ALTER TABLE account\_type ADD CONSTRAINT account\_type\_pk PRIMARY KEY ( id );

CREATE TABLE artist (

    id          INTEGER NOT NULL,

    name        VARCHAR2(20) NOT NULL,

    last\_name   VARCHAR2(20) NOT NULL,

    birth       DATE NOT NULL,

    country\_id  INTEGER NOT NULL,

    gender\_id   INTEGER NOT NULL

);

ALTER TABLE artist ADD CONSTRAINT artist\_pk PRIMARY KEY ( id );

CREATE TABLE based (

    id    INTEGER NOT NULL,

    name  VARCHAR2(20) NOT NULL

);

ALTER TABLE based ADD CONSTRAINT based\_pk PRIMARY KEY ( id );

CREATE TABLE cast\_documentary (

    id              INTEGER NOT NULL,

    artist\_id       INTEGER NOT NULL,

    documentary\_id  INTEGER NOT NULL

);

ALTER TABLE cast\_documentary ADD CONSTRAINT cast\_documentary\_pk PRIMARY KEY ( id );

CREATE TABLE cast\_documentary\_role (

    cast\_documentary\_id  INTEGER NOT NULL,

    role\_id              INTEGER NOT NULL

);

ALTER TABLE cast\_documentary\_role ADD CONSTRAINT cast\_documentary\_role\_pk PRIMARY KEY ( cast\_documentary\_id,

                                                                                        role\_id );

CREATE TABLE cast\_movie (

    id         INTEGER NOT NULL,

    artist\_id  INTEGER NOT NULL,

    movie\_id   INTEGER NOT NULL

);

ALTER TABLE cast\_movie ADD CONSTRAINT cast\_movie\_pk PRIMARY KEY ( id );

CREATE TABLE cast\_movie\_role (

    cast\_movie\_id  INTEGER NOT NULL,

    role\_id        INTEGER NOT NULL

);

ALTER TABLE cast\_movie\_role ADD CONSTRAINT cast\_movie\_role\_pk PRIMARY KEY ( cast\_movie\_id,

                                                                            role\_id );

CREATE TABLE cast\_serie (

    id         INTEGER NOT NULL,

    artist\_id  INTEGER NOT NULL,

    serie\_id   INTEGER NOT NULL

);

ALTER TABLE cast\_serie ADD CONSTRAINT cast\_serie\_pk PRIMARY KEY ( id );

CREATE TABLE cast\_serie\_role (

    cast\_serie\_id  INTEGER NOT NULL,

    role\_id        INTEGER NOT NULL

);

ALTER TABLE cast\_serie\_role ADD CONSTRAINT cast\_serie\_role\_pk PRIMARY KEY ( role\_id,

                                                                            cast\_serie\_id );

CREATE TABLE client (

    id               INTEGER NOT NULL,

    name             VARCHAR2(30) NOT NULL,

    last\_name        VARCHAR2(30) NOT NULL,

    direction        VARCHAR2(20) NOT NULL,

    phone            VARCHAR2(15) NOT NULL,

    birth            DATE NOT NULL,

    email            VARCHAR2(60) NOT NULL,

    inclusion        DATE NOT NULL,

    country\_id       INTEGER NOT NULL,

    account\_type\_id  INTEGER NOT NULL

);

ALTER TABLE client ADD CONSTRAINT client\_pk PRIMARY KEY ( id );

CREATE TABLE country (

    id    INTEGER NOT NULL,

    name  VARCHAR2(20) NOT NULL

);

ALTER TABLE country ADD CONSTRAINT country\_pk PRIMARY KEY ( id );

CREATE TABLE device (

    id         INTEGER NOT NULL,

    name       VARCHAR2(45) NOT NULL,

    client\_id  INTEGER NOT NULL

);

ALTER TABLE device ADD CONSTRAINT device\_pk PRIMARY KEY ( id );

CREATE TABLE documentary (

    id            INTEGER NOT NULL,

    title         VARCHAR2(50) NOT NULL,

    abstract      VARCHAR2(70) NOT NULL,

    begin\_date    DATE NOT NULL,

    length        INTEGER NOT NULL,

    seasons       INTEGER NOT NULL,

    chapters      INTEGER NOT NULL,

    rated\_id      INTEGER NOT NULL,

    docuserie\_id  INTEGER

);

ALTER TABLE documentary ADD CONSTRAINT documentary\_pk PRIMARY KEY ( id );

CREATE TABLE documentary\_format (

    format\_id       INTEGER NOT NULL,

    documentary\_id  INTEGER NOT NULL

);

ALTER TABLE documentary\_format ADD CONSTRAINT documentary\_format\_pk PRIMARY KEY ( format\_id,

                                                                                  documentary\_id );

CREATE TABLE documentary\_genre (

    genre\_id        INTEGER NOT NULL,

    documentary\_id  INTEGER NOT NULL

);

ALTER TABLE documentary\_genre ADD CONSTRAINT documentary\_genre\_pk PRIMARY KEY ( documentary\_id,

                                                                                genre\_id );

CREATE TABLE documentary\_language (

    id              INTEGER NOT NULL,

    audio\_id        INTEGER NOT NULL,

    subtitles\_id    INTEGER,

    documentary\_id  INTEGER NOT NULL

);

ALTER TABLE documentary\_language ADD CONSTRAINT documentary\_language\_pk PRIMARY KEY ( id );

CREATE TABLE docuserie (

    id        INTEGER NOT NULL,

    title     VARCHAR2(50) NOT NULL,

    abstract  VARCHAR2(70) NOT NULL,

    based\_id  INTEGER NOT NULL

);

ALTER TABLE docuserie ADD CONSTRAINT docuserie\_pk PRIMARY KEY ( id );

CREATE TABLE download\_documentary (

    id                       INTEGER NOT NULL,

    begin\_date               DATE NOT NULL,

    recent\_minute            INTEGER NOT NULL,

    client\_id                INTEGER NOT NULL,

    state\_id                 INTEGER NOT NULL,

    documentary\_language\_id  INTEGER NOT NULL

);

ALTER TABLE download\_documentary ADD CONSTRAINT download\_documentary\_pk PRIMARY KEY ( id );

CREATE TABLE download\_movie (

    id                 INTEGER NOT NULL,

    begin\_date         DATE NOT NULL,

    recent\_minute      INTEGER NOT NULL,

    client\_id          INTEGER NOT NULL,

    state\_id           INTEGER NOT NULL,

    movie\_language\_id  INTEGER NOT NULL

);

ALTER TABLE download\_movie ADD CONSTRAINT download\_movie\_pk PRIMARY KEY ( id );

CREATE TABLE download\_serie (

    id                 INTEGER NOT NULL,

    begin\_date         DATE NOT NULL,

    recent\_minute      INTEGER NOT NULL,

    client\_id          INTEGER NOT NULL,

    state\_id           INTEGER NOT NULL,

    serie\_language\_id  INTEGER NOT NULL

);

ALTER TABLE download\_serie ADD CONSTRAINT download\_serie\_pk PRIMARY KEY ( id );

CREATE TABLE format (

    id    INTEGER NOT NULL,

    type  VARCHAR2(20) NOT NULL

);

ALTER TABLE format ADD CONSTRAINT format\_pk PRIMARY KEY ( id );

CREATE TABLE gender (

    id    INTEGER NOT NULL,

    name  VARCHAR2(20) NOT NULL

);

ALTER TABLE gender ADD CONSTRAINT gender\_pk PRIMARY KEY ( id );

CREATE TABLE genre (

    id    INTEGER NOT NULL,

    name  VARCHAR2(20) NOT NULL

);

ALTER TABLE genre ADD CONSTRAINT genre\_pk PRIMARY KEY ( id );

CREATE TABLE language (

    id    INTEGER NOT NULL,

    name  VARCHAR2(20) NOT NULL

);

ALTER TABLE language ADD CONSTRAINT language\_pk PRIMARY KEY ( id );

CREATE TABLE macroserie (

    id        INTEGER NOT NULL,

    title     VARCHAR2(50) NOT NULL,

    abstract  VARCHAR2(70) NOT NULL,

    based\_id  INTEGER NOT NULL

);

ALTER TABLE macroserie ADD CONSTRAINT macroserie\_pk PRIMARY KEY ( id );

CREATE TABLE movie (

    id          INTEGER NOT NULL,

    title       VARCHAR2(50) NOT NULL,

    year        DATE NOT NULL,

    length      INTEGER NOT NULL,

    country\_id  INTEGER NOT NULL,

    studio\_id   INTEGER NOT NULL,

    rated\_id    INTEGER NOT NULL,

    saga\_id     INTEGER

);

ALTER TABLE movie ADD CONSTRAINT movie\_pk PRIMARY KEY ( id );

CREATE TABLE movie\_format (

    format\_id  INTEGER NOT NULL,

    movie\_id   INTEGER NOT NULL

);

ALTER TABLE movie\_format ADD CONSTRAINT movie\_format\_pk PRIMARY KEY ( format\_id,

                                                                      movie\_id );

CREATE TABLE movie\_genre (

    genre\_id  INTEGER NOT NULL,

    movie\_id  INTEGER NOT NULL

);

ALTER TABLE movie\_genre ADD CONSTRAINT movie\_genre\_pk PRIMARY KEY ( genre\_id,

                                                                    movie\_id );

CREATE TABLE movie\_language (

    id            INTEGER NOT NULL,

    audio\_id      INTEGER NOT NULL,

    subtitles\_id  INTEGER,

    movie\_id      INTEGER NOT NULL

);

ALTER TABLE movie\_language ADD CONSTRAINT movie\_language\_pk PRIMARY KEY ( id );

CREATE TABLE rated (

    id    INTEGER NOT NULL,

    name  VARCHAR2(20) NOT NULL

);

ALTER TABLE rated ADD CONSTRAINT rated\_pk PRIMARY KEY ( id );

CREATE TABLE reproduction\_documentary (

    id                       INTEGER NOT NULL,

    begin\_date               DATE NOT NULL,

    recent\_minute            INTEGER NOT NULL,

    client\_id                INTEGER NOT NULL,

    state\_id                 INTEGER NOT NULL,

    documentary\_language\_id  INTEGER NOT NULL

);

ALTER TABLE reproduction\_documentary ADD CONSTRAINT reproduction\_documentary\_pk PRIMARY KEY ( id );

CREATE TABLE reproduction\_movie (

    id                 INTEGER NOT NULL,

    begin\_date         DATE NOT NULL,

    recent\_minute      INTEGER NOT NULL,

    client\_id          INTEGER NOT NULL,

    state\_id           INTEGER NOT NULL,

    movie\_language\_id  INTEGER NOT NULL

);

ALTER TABLE reproduction\_movie ADD CONSTRAINT reproduction\_movie\_pk PRIMARY KEY ( id );

CREATE TABLE reproduction\_serie (

    id                 INTEGER NOT NULL,

    begin\_date         DATE NOT NULL,

    recent\_minute      INTEGER NOT NULL,

    client\_id          INTEGER NOT NULL,

    state\_id           INTEGER NOT NULL,

    serie\_language\_id  INTEGER NOT NULL

);

ALTER TABLE reproduction\_serie ADD CONSTRAINT reproduction\_serie\_pk PRIMARY KEY ( id );

CREATE TABLE role (

    id    INTEGER NOT NULL,

    name  VARCHAR2(20) NOT NULL

);

ALTER TABLE role ADD CONSTRAINT role\_pk PRIMARY KEY ( id );

CREATE TABLE saga (

    id        INTEGER NOT NULL,

    title     VARCHAR2(50) NOT NULL,

    abstract  VARCHAR2(70) NOT NULL,

    based\_id  INTEGER NOT NULL

);

ALTER TABLE saga ADD CONSTRAINT saga\_pk PRIMARY KEY ( id );

CREATE TABLE serie (

    id             INTEGER NOT NULL,

    title          VARCHAR2(50) NOT NULL,

    abstract       VARCHAR2(70) NOT NULL,

    begin\_date     DATE NOT NULL,

    length         INTEGER NOT NULL,

    seasons        INTEGER NOT NULL,

    chapters       INTEGER NOT NULL,

    rated\_id       INTEGER NOT NULL,

    macroserie\_id  INTEGER,

    country\_id     INTEGER NOT NULL

);

ALTER TABLE serie ADD CONSTRAINT serie\_pk PRIMARY KEY ( id );

CREATE TABLE serie\_format (

    format\_id  INTEGER NOT NULL,

    serie\_id   INTEGER NOT NULL

);

ALTER TABLE serie\_format ADD CONSTRAINT serie\_format\_pk PRIMARY KEY ( format\_id,

                                                                      serie\_id );

CREATE TABLE serie\_genre (

    genre\_id  INTEGER NOT NULL,

    serie\_id  INTEGER NOT NULL

);

ALTER TABLE serie\_genre ADD CONSTRAINT serie\_genre\_pk PRIMARY KEY ( genre\_id,

                                                                    serie\_id );

CREATE TABLE serie\_language (

    id            INTEGER NOT NULL,

    audio\_id      INTEGER NOT NULL,

    subtitles\_id  INTEGER,

    serie\_id      INTEGER NOT NULL

);

ALTER TABLE serie\_language ADD CONSTRAINT serie\_language\_pk PRIMARY KEY ( id );

CREATE TABLE state (

    id     INTEGER NOT NULL,

    state  VARCHAR2(20) NOT NULL

);

ALTER TABLE state ADD CONSTRAINT state\_pk PRIMARY KEY ( id );

CREATE TABLE studio (

    id    INTEGER NOT NULL,

    name  VARCHAR2(30) NOT NULL

);

ALTER TABLE studio ADD CONSTRAINT studio\_pk PRIMARY KEY ( id );

ALTER TABLE artist

    ADD CONSTRAINT artist\_country\_fk FOREIGN KEY ( country\_id )

        REFERENCES country ( id );

ALTER TABLE artist

    ADD CONSTRAINT artist\_gender\_fk FOREIGN KEY ( gender\_id )

        REFERENCES gender ( id );

ALTER TABLE cast\_documentary

    ADD CONSTRAINT cast\_documentary\_artist\_fk FOREIGN KEY ( artist\_id )

        REFERENCES artist ( id );

ALTER TABLE cast\_documentary

    ADD CONSTRAINT cast\_documentary\_fk FOREIGN KEY ( documentary\_id )

        REFERENCES documentary ( id );

ALTER TABLE cast\_documentary\_role

    ADD CONSTRAINT cast\_documentary\_role\_fk FOREIGN KEY ( cast\_documentary\_id )

        REFERENCES cast\_documentary ( id );

ALTER TABLE cast\_documentary\_role

    ADD CONSTRAINT cast\_documentary\_role\_role\_fk FOREIGN KEY ( role\_id )

        REFERENCES role ( id );

ALTER TABLE cast\_movie

    ADD CONSTRAINT cast\_movie\_artist\_fk FOREIGN KEY ( artist\_id )

        REFERENCES artist ( id );

ALTER TABLE cast\_movie

    ADD CONSTRAINT cast\_movie\_movie\_fk FOREIGN KEY ( movie\_id )

        REFERENCES movie ( id );

ALTER TABLE cast\_movie\_role

    ADD CONSTRAINT cast\_movie\_role\_movie\_fk FOREIGN KEY ( cast\_movie\_id )

        REFERENCES cast\_movie ( id );

ALTER TABLE cast\_movie\_role

    ADD CONSTRAINT cast\_movie\_role\_role\_fk FOREIGN KEY ( role\_id )

        REFERENCES role ( id );

ALTER TABLE cast\_serie

    ADD CONSTRAINT cast\_serie\_artist\_fk FOREIGN KEY ( artist\_id )

        REFERENCES artist ( id );

ALTER TABLE cast\_serie\_role

    ADD CONSTRAINT cast\_serie\_role\_cast\_fk FOREIGN KEY ( cast\_serie\_id )

        REFERENCES cast\_serie ( id );

ALTER TABLE cast\_serie\_role

    ADD CONSTRAINT cast\_serie\_role\_role\_fk FOREIGN KEY ( role\_id )

        REFERENCES role ( id );

ALTER TABLE cast\_serie

    ADD CONSTRAINT cast\_serie\_serie\_fk FOREIGN KEY ( serie\_id )

        REFERENCES serie ( id );

ALTER TABLE client

    ADD CONSTRAINT client\_account\_type\_fk FOREIGN KEY ( account\_type\_id )

        REFERENCES account\_type ( id );

ALTER TABLE client

    ADD CONSTRAINT client\_country\_fk FOREIGN KEY ( country\_id )

        REFERENCES country ( id );

ALTER TABLE device

    ADD CONSTRAINT device\_client\_fk FOREIGN KEY ( client\_id )

        REFERENCES client ( id );

ALTER TABLE documentary

    ADD CONSTRAINT documentary\_docuserie\_fk FOREIGN KEY ( docuserie\_id )

        REFERENCES docuserie ( id );

ALTER TABLE documentary\_format

    ADD CONSTRAINT documentary\_format\_fk FOREIGN KEY ( documentary\_id )

        REFERENCES documentary ( id );

ALTER TABLE documentary\_format

    ADD CONSTRAINT documentary\_format\_format\_fk FOREIGN KEY ( format\_id )

        REFERENCES format ( id );

ALTER TABLE documentary\_genre

    ADD CONSTRAINT documentary\_genre\_docu\_fk FOREIGN KEY ( documentary\_id )

        REFERENCES documentary ( id );

ALTER TABLE documentary\_genre

    ADD CONSTRAINT documentary\_genre\_genre\_fk FOREIGN KEY ( genre\_id )

        REFERENCES genre ( id );

ALTER TABLE documentary\_language

    ADD CONSTRAINT documentary\_lang\_fk FOREIGN KEY ( documentary\_id )

        REFERENCES documentary ( id );

ALTER TABLE documentary\_language

    ADD CONSTRAINT documentary\_lang\_lang\_fk FOREIGN KEY ( audio\_id )

        REFERENCES language ( id );

ALTER TABLE documentary\_language

    ADD CONSTRAINT documentary\_lang\_lang\_fkv2 FOREIGN KEY ( subtitles\_id )

        REFERENCES language ( id );

ALTER TABLE documentary

    ADD CONSTRAINT documentary\_rated\_fk FOREIGN KEY ( rated\_id )

        REFERENCES rated ( id );

ALTER TABLE docuserie

    ADD CONSTRAINT docuserie\_based\_fk FOREIGN KEY ( based\_id )

        REFERENCES based ( id );

ALTER TABLE download\_documentary

    ADD CONSTRAINT download\_documentary\_client\_fk FOREIGN KEY ( client\_id )

        REFERENCES client ( id );

ALTER TABLE download\_documentary

    ADD CONSTRAINT download\_documentary\_lang\_fk FOREIGN KEY ( documentary\_language\_id )

        REFERENCES documentary\_language ( id );

ALTER TABLE download\_documentary

    ADD CONSTRAINT download\_documentary\_state\_fk FOREIGN KEY ( state\_id )

        REFERENCES state ( id );

ALTER TABLE download\_movie

    ADD CONSTRAINT download\_movie\_client\_fk FOREIGN KEY ( client\_id )

        REFERENCES client ( id );

ALTER TABLE download\_movie

    ADD CONSTRAINT download\_movie\_lang\_fk FOREIGN KEY ( movie\_language\_id )

        REFERENCES movie\_language ( id );

ALTER TABLE download\_movie

    ADD CONSTRAINT download\_movie\_state\_fk FOREIGN KEY ( state\_id )

        REFERENCES state ( id );

ALTER TABLE download\_serie

    ADD CONSTRAINT download\_serie\_client\_fk FOREIGN KEY ( client\_id )

        REFERENCES client ( id );

ALTER TABLE download\_serie

    ADD CONSTRAINT download\_serie\_lang\_fk FOREIGN KEY ( serie\_language\_id )

        REFERENCES serie\_language ( id );

ALTER TABLE download\_serie

    ADD CONSTRAINT download\_serie\_state\_fk FOREIGN KEY ( state\_id )

        REFERENCES state ( id );

ALTER TABLE macroserie

    ADD CONSTRAINT macroserie\_based\_fk FOREIGN KEY ( based\_id )

        REFERENCES based ( id );

ALTER TABLE movie

    ADD CONSTRAINT movie\_country\_fk FOREIGN KEY ( country\_id )

        REFERENCES country ( id );

ALTER TABLE movie\_format

    ADD CONSTRAINT movie\_format\_format\_fk FOREIGN KEY ( format\_id )

        REFERENCES format ( id );

ALTER TABLE movie\_format

    ADD CONSTRAINT movie\_format\_movie\_fk FOREIGN KEY ( movie\_id )

        REFERENCES movie ( id );

ALTER TABLE movie\_genre

    ADD CONSTRAINT movie\_genre\_genre\_fk FOREIGN KEY ( genre\_id )

        REFERENCES genre ( id );

ALTER TABLE movie\_genre

    ADD CONSTRAINT movie\_genre\_movie\_fk FOREIGN KEY ( movie\_id )

        REFERENCES movie ( id );

ALTER TABLE movie\_language

    ADD CONSTRAINT movie\_lang\_lang\_fk FOREIGN KEY ( subtitles\_id )

        REFERENCES language ( id );

ALTER TABLE movie\_language

    ADD CONSTRAINT movie\_lang\_lang\_fkv2 FOREIGN KEY ( audio\_id )

        REFERENCES language ( id );

ALTER TABLE movie\_language

    ADD CONSTRAINT movie\_lang\_movie\_fk FOREIGN KEY ( movie\_id )

        REFERENCES movie ( id );

ALTER TABLE movie

    ADD CONSTRAINT movie\_rated\_fk FOREIGN KEY ( rated\_id )

        REFERENCES rated ( id );

ALTER TABLE movie

    ADD CONSTRAINT movie\_saga\_fk FOREIGN KEY ( saga\_id )

        REFERENCES saga ( id );

ALTER TABLE movie

    ADD CONSTRAINT movie\_studio\_fk FOREIGN KEY ( studio\_id )

        REFERENCES studio ( id );

ALTER TABLE reproduction\_documentary

    ADD CONSTRAINT reproduction\_docu\_client\_fk FOREIGN KEY ( client\_id )

        REFERENCES client ( id );

ALTER TABLE reproduction\_documentary

    ADD CONSTRAINT reproduction\_docu\_lang\_fk FOREIGN KEY ( documentary\_language\_id )

        REFERENCES documentary\_language ( id );

ALTER TABLE reproduction\_documentary

    ADD CONSTRAINT reproduction\_docu\_state\_fk FOREIGN KEY ( state\_id )

        REFERENCES state ( id );

ALTER TABLE reproduction\_movie

    ADD CONSTRAINT reproduction\_movie\_client\_fk FOREIGN KEY ( client\_id )

        REFERENCES client ( id );

ALTER TABLE reproduction\_movie

    ADD CONSTRAINT reproduction\_movie\_lang\_fk FOREIGN KEY ( movie\_language\_id )

        REFERENCES movie\_language ( id );

ALTER TABLE reproduction\_movie

    ADD CONSTRAINT reproduction\_movie\_state\_fk FOREIGN KEY ( state\_id )

        REFERENCES state ( id );

ALTER TABLE reproduction\_serie

    ADD CONSTRAINT reproduction\_serie\_client\_fk FOREIGN KEY ( client\_id )

        REFERENCES client ( id );

ALTER TABLE reproduction\_serie

    ADD CONSTRAINT reproduction\_serie\_lang\_fk FOREIGN KEY ( serie\_language\_id )

        REFERENCES serie\_language ( id );

ALTER TABLE reproduction\_serie

    ADD CONSTRAINT reproduction\_serie\_state\_fk FOREIGN KEY ( state\_id )

        REFERENCES state ( id );

ALTER TABLE saga

    ADD CONSTRAINT saga\_based\_fk FOREIGN KEY ( based\_id )

        REFERENCES based ( id );

ALTER TABLE serie

    ADD CONSTRAINT serie\_country\_fk FOREIGN KEY ( country\_id )

        REFERENCES country ( id );

ALTER TABLE serie\_format

    ADD CONSTRAINT serie\_format\_format\_fk FOREIGN KEY ( format\_id )

        REFERENCES format ( id );

ALTER TABLE serie\_format

    ADD CONSTRAINT serie\_format\_serie\_fk FOREIGN KEY ( serie\_id )

        REFERENCES serie ( id );

ALTER TABLE serie\_genre

    ADD CONSTRAINT serie\_genre\_genre\_fk FOREIGN KEY ( genre\_id )

        REFERENCES genre ( id );

ALTER TABLE serie\_genre

    ADD CONSTRAINT serie\_genre\_serie\_fk FOREIGN KEY ( serie\_id )

        REFERENCES serie ( id );

ALTER TABLE serie\_language

    ADD CONSTRAINT serie\_lang\_lang\_fk FOREIGN KEY ( audio\_id )

        REFERENCES language ( id );

ALTER TABLE serie\_language

    ADD CONSTRAINT serie\_lang\_lang\_fkv2 FOREIGN KEY ( subtitles\_id )

        REFERENCES language ( id );

ALTER TABLE serie\_language

    ADD CONSTRAINT serie\_lang\_serie\_fk FOREIGN KEY ( serie\_id )

        REFERENCES serie ( id );

ALTER TABLE serie

    ADD CONSTRAINT serie\_macroserie\_fk FOREIGN KEY ( macroserie\_id )

        REFERENCES macroserie ( id );

ALTER TABLE serie

    ADD CONSTRAINT serie\_rated\_fk FOREIGN KEY ( rated\_id )

        REFERENCES rated ( id );

-- Oracle SQL Developer Data Modeler Summary Report:

--

-- CREATE TABLE                            41

-- CREATE INDEX                             0

-- ALTER TABLE                            109

-- CREATE VIEW                              0

-- ALTER VIEW                               0

-- CREATE PACKAGE                           0

-- CREATE PACKAGE BODY                      0

-- CREATE PROCEDURE                         0

-- CREATE FUNCTION                          0

-- CREATE TRIGGER                           0

-- ALTER TRIGGER                            0

-- CREATE COLLECTION TYPE                   0

-- CREATE STRUCTURED TYPE                   0

-- CREATE STRUCTURED TYPE BODY              0

-- CREATE CLUSTER                           0

-- CREATE CONTEXT                           0

-- CREATE DATABASE                          0

-- CREATE DIMENSION                         0

-- CREATE DIRECTORY                         0

-- CREATE DISK GROUP                        0

-- CREATE ROLE                              0

-- CREATE ROLLBACK SEGMENT                  0

-- CREATE SEQUENCE                          0

-- CREATE MATERIALIZED VIEW                 0

-- CREATE MATERIALIZED VIEW LOG             0

-- CREATE SYNONYM                           0

-- CREATE TABLESPACE                        0

-- CREATE USER                              0

--

-- DROP TABLESPACE                          0

-- DROP DATABASE                            0

--

-- REDACTION POLICY                         0

--

-- ORDS DROP SCHEMA                         0

-- ORDS ENABLE SCHEMA                       0

-- ORDS ENABLE OBJECT                       0

--

-- ERRORS                                   0

-- WARNINGS                                 0

Reportes

Reporte #1

SELECT \* FROM (

  SELECT 'Pelicula' Tipo, m.title Titulo,

  EXTRACT(YEAR FROM m.year) Lanzamiento,

  c.name Pais,

  (SELECT LISTAGG(g.name, ', ') WITHIN GROUP (ORDER BY g.name) FROM genre g

    INNER JOIN movie\_genre mg ON mg.genre\_id = g.id

    WHERE mg.movie\_id = m.id

  ) Generos,

  (SELECT LISTAGG(a.name || ' ' || a.last\_name, ', ') WITHIN GROUP (ORDER BY a.name || ' ' || a.last\_name) FROM artist a

    INNER JOIN cast\_movie cm ON cm.artist\_id = a.id

    INNER JOIN cast\_movie\_role cmr ON cmr.cast\_movie\_id = cm.id

    WHERE cm.movie\_id = m.id AND cmr.role\_id = 3

  ) Directores

  FROM movie m

  INNER JOIN country c ON c.id = m.country\_id

  UNION

  SELECT 'Serie' Tipo, s.title Titulo,

  EXTRACT(YEAR FROM s.begin\_date) Lanzamiento,

  c.name Pais,

  (SELECT LISTAGG(g.name, ', ') WITHIN GROUP (ORDER BY g.name) FROM genre g

    INNER JOIN serie\_genre sg ON sg.genre\_id = g.id

    WHERE sg.serie\_id = s.id

  ) Generos,

  (SELECT LISTAGG(a.name || ' ' || a.last\_name, ', ') WITHIN GROUP (ORDER BY a.name || ' ' || a.last\_name) FROM artist a

    INNER JOIN cast\_serie cs ON cs.artist\_id = a.id

    INNER JOIN cast\_serie\_role csr ON csr.cast\_serie\_id = cs.id

    WHERE cs.serie\_id = s.id AND csr.role\_id = 3

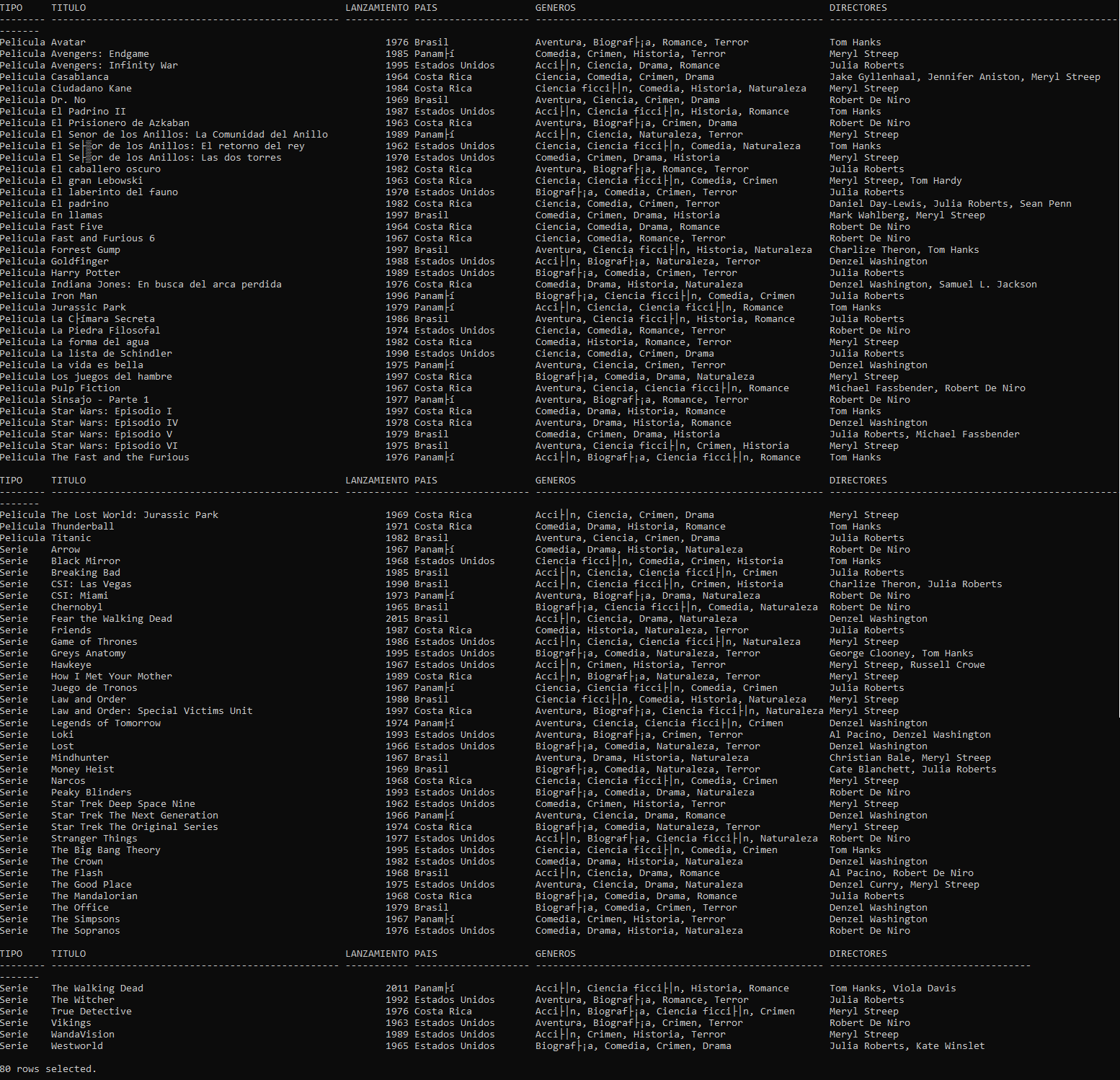
  ) Directores

  FROM serie s

  INNER JOIN country c ON c.id = s.country\_id

);

Resultado:



Reporte #2

SELECT \* FROM (

  SELECT 'Serie' Tipo, s.title Titulo,

  EXTRACT(YEAR FROM s.begin\_date) Lanzamiento,

  s.seasons Temporadas,

  (SELECT LISTAGG(g.name, ', ') WITHIN GROUP (ORDER BY g.name) FROM genre g

    INNER JOIN serie\_genre sg ON sg.genre\_id = g.id

    WHERE sg.serie\_id = s.id

  ) Generos,

  (SELECT LISTAGG(a.name || ' ' || a.last\_name, ', ') WITHIN GROUP (ORDER BY a.name || ' ' || a.last\_name) FROM artist a

    INNER JOIN cast\_serie cs ON cs.artist\_id = a.id

    INNER JOIN cast\_serie\_role csr ON csr.cast\_serie\_id = cs.id

    WHERE cs.serie\_id = s.id AND csr.role\_id = 3

  ) Directores

  FROM serie s

  UNION

  SELECT 'Documental' Tipo, d.title Titulo,

  EXTRACT(YEAR FROM d.begin\_date) Lanzamiento,

  d.seasons Temporadas,

  (SELECT LISTAGG(g.name, ', ') WITHIN GROUP (ORDER BY g.name) FROM genre g

    INNER JOIN documentary\_genre dg ON dg.genre\_id = g.id

    WHERE dg.documentary\_id = d.id

  ) Generos,

  (SELECT LISTAGG(a.name || ' ' || a.last\_name, ', ') WITHIN GROUP (ORDER BY a.name || ' ' || a.last\_name) FROM artist a

    INNER JOIN cast\_documentary cd ON cd.artist\_id = a.id

    INNER JOIN cast\_documentary\_role cdr ON cdr.cast\_documentary\_id = cd.id

    WHERE cd.documentary\_id = d.id AND cdr.role\_id = 3

  ) Directores

  FROM documentary d

);

Resultado:



Reporte #3

SELECT c.name Nombre, c.last\_name Apellidos, COUNT(DISTINCT rd.id) Documentales, COUNT(DISTINCT rm.id) Peliculas, COUNT(DISTINCT rs.id) Series

FROM client c

INNER JOIN reproduction\_documentary rd ON rd.client\_id = c.id

INNER JOIN reproduction\_movie rm ON rm.client\_id = c.id

INNER JOIN reproduction\_serie rs ON rs.client\_id = c.id

GROUP BY c.name, c.last\_name;

Resultado:

A screen shot of a computer

Description automatically generated

Reporte #4

WITH movie\_genres\_cte AS (

  SELECT mg.movie\_id, LISTAGG(g.name, ', ') WITHIN GROUP (ORDER BY g.name) AS generos

  FROM movie\_genre mg

  INNER JOIN genre g ON g.id = mg.genre\_id

  GROUP BY mg.movie\_id

)

SELECT

  m.title AS Titulo,

  EXTRACT(YEAR FROM m.year) AS Lanzamiento,

  mgc.generos AS Generos,

  COUNT(DISTINCT rm.id) AS Reproducciones

FROM movie m

INNER JOIN movie\_language ml ON ml.movie\_id = m.id

INNER JOIN reproduction\_movie rm ON rm.movie\_language\_id = ml.id

INNER JOIN client c ON c.id = rm.client\_id

LEFT JOIN movie\_genres\_cte mgc ON mgc.movie\_id = m.id

GROUP BY m.title, mgc.generos, EXTRACT(YEAR FROM m.year);

Resultado:

A screen shot of a computer screen

Description automatically generated

Reporte #5

SELECT s.title Titulo, EXTRACT(YEAR FROM s.begin\_date) Lanzamiento, s.seasons Temporadas, COUNT(DISTINCT rs.id) Reproducciones

FROM serie s

INNER JOIN serie\_language sl ON sl.serie\_id = s.id

INNER JOIN reproduction\_serie rs ON rs.serie\_language\_id = sl.id

INNER JOIN client c ON c.id = rs.client\_id

GROUP BY s.title, s.seasons, s.begin\_date

ORDER BY Lanzamiento, Titulo;

Resultado:

A screenshot of a computer

Description automatically generated

Reporte #6

SELECT d.title Titulo, EXTRACT(YEAR FROM d.begin\_date) Lanzamiento, d.seasons Temporadas, COUNT(DISTINCT rd.id) Reproducciones

FROM documentary d

INNER JOIN documentary\_language dl ON dl.documentary\_id = d.id

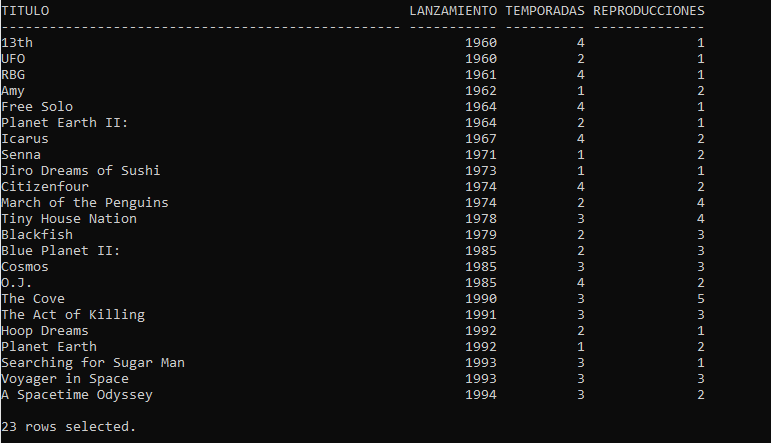
INNER JOIN reproduction\_documentary rd ON rd.documentary\_language\_id = dl.id

INNER JOIN client c ON c.id = rd.client\_id

GROUP BY d.title, d.seasons, d.begin\_date

ORDER BY Lanzamiento, Titulo;

Resultado:



Reporte #7

SELECT \* FROM (

  SELECT 'Pelicula' Tipo, m.title Titulo, EXTRACT(YEAR FROM m.year) Lazamiento, m.length Duracion, COUNT(1) Descargas, MAX(dm.begin\_date) "ULT. DESC."

  FROM download\_movie dm

  INNER JOIN movie\_language ml ON ml.id = dm.movie\_language\_id

  INNER JOIN movie m ON m.id = ml.movie\_id

  GROUP BY m.title, m.year, m.length

  UNION

  SELECT 'Serie' Tipo, s.title Titulo, EXTRACT(YEAR FROM s.begin\_date) Lazamiento, s.length Duracion, COUNT(1) Descargas, MAX(ds.begin\_date) "ULT. DESC."

  FROM download\_serie ds

  INNER JOIN serie\_language sl ON sl.id = ds.serie\_language\_id

  INNER JOIN serie s ON s.id = sl.serie\_id

  GROUP BY s.title, s.begin\_date, s.length

  UNION

  SELECT 'Documental' Tipo, d.title Titulo, EXTRACT(YEAR FROM d.begin\_date) Lazamiento, d.length Duracion, COUNT(1) Descargas, MAX(dd.begin\_date) "ULT. DESC."

  FROM download\_documentary dd

  INNER JOIN documentary\_language dl ON dl.id = dd.documentary\_language\_id

  INNER JOIN documentary d ON d.id = dl.documentary\_id

  GROUP BY d.title, d.begin\_date, d.length

)

ORDER BY Descargas DESC;

Resultado:

A screenshot of a computer screen

Description automatically generated

Reporte #8

SELECT name || ' ' || last\_name "NOMBRE COMPLETO",

EXTRACT(YEAR FROM inclusion) Suscripcion,

SUM(repro) "PELICULAS REPRODUCIDAS",

SUM(downloads) "TOTAL DESC.",

last\_movie\_title "TITULO ULT. PELICULA.",

MAX(last\_reproduction) "ULT. REPR."

FROM (

  SELECT c.id, c.name, c.last\_name, c.inclusion,

  COUNT(DISTINCT CASE WHEN dm.client\_id = c.id THEN dm.id END) downloads,

  COUNT(DISTINCT CASE WHEN rm.client\_id = c.id THEN rm.id END) repro,

  MAX(rm.begin\_date) last\_reproduction

  FROM client c

  LEFT JOIN download\_movie dm ON dm.client\_id = c.id

  LEFT JOIN reproduction\_movie rm ON rm.client\_id = c.id

  GROUP BY c.id, c.name, c.last\_name, c.inclusion

  UNION

  SELECT c.id, c.name, c.last\_name, c.inclusion,

  COUNT(DISTINCT CASE WHEN ds.client\_id = c.id THEN ds.id END) downloads, 0,

  MAX(rs.begin\_date) last\_reproduction

  FROM client c

  LEFT JOIN download\_serie ds ON ds.client\_id = c.id

  LEFT JOIN reproduction\_serie rs ON rs.client\_id = c.id

  GROUP BY c.id, c.name, c.last\_name, c.inclusion

  UNION

  SELECT c.id, c.name, c.last\_name, c.inclusion,

  COUNT(DISTINCT CASE WHEN dd.client\_id = c.id THEN dd.id END) downloads, 0,

  MAX(rd.begin\_date) last\_reproduction

  FROM client c

  LEFT JOIN download\_documentary dd ON dd.client\_id = c.id

  LEFT JOIN reproduction\_documentary rd ON rd.client\_id = c.id

  GROUP BY c.id, c.name, c.last\_name, c.inclusion

) t

LEFT JOIN (

  SELECT rm.client\_id, m.title AS last\_movie\_title

  FROM reproduction\_movie rm

  INNER JOIN movie\_language ml ON ml.id = rm.movie\_language\_id

  INNER JOIN movie m ON m.id = ml.movie\_id

  WHERE rm.begin\_date = (

    SELECT MAX(rm\_inner.begin\_date)

    FROM reproduction\_movie rm\_inner

    WHERE rm\_inner.client\_id = rm.client\_id

  )

) r ON r.client\_id = t.id

GROUP BY name, last\_name, EXTRACT(YEAR FROM inclusion), last\_movie\_title

ORDER BY last\_name;

Resultado:

A screen shot of a computer

Description automatically generated

Reporte #9

SELECT ROWNUM Top, r.\* FROM (

  SELECT Titulo, Tipo, Reproducciones, Descargas, last\_reproduction "ULT. REPR.", last\_download "ULT. DESC." FROM (

    SELECT 'Pelicula' Tipo, m.title Titulo,

      COUNT(DISTINCT dm.id) Descargas,

      COUNT(DISTINCT rm.id) Reproducciones,

      MAX(rm.begin\_date) last\_reproduction,

      MAX(dm.begin\_date) last\_download,

      COUNT(DISTINCT dm.id) + COUNT(DISTINCT rm.id) Total

      FROM movie m

      LEFT JOIN movie\_language ml ON ml.movie\_id = m.id

      LEFT JOIN download\_movie dm ON dm.movie\_language\_id = ml.id

      LEFT JOIN reproduction\_movie rm ON rm.movie\_language\_id = ml.id

      HAVING COUNT(DISTINCT dm.id) > 0 OR COUNT(DISTINCT rm.id) > 0

      GROUP BY m.title

    UNION

    SELECT 'Serie' Tipo, s.title Titulo,

      COUNT(DISTINCT ds.id) Descargas,

      COUNT(DISTINCT rs.id) Reproducciones,

      MAX(rs.begin\_date) last\_reproduction,

      MAX(ds.begin\_date) last\_download,

      COUNT(DISTINCT ds.id) + COUNT(DISTINCT rs.id) Total

      FROM serie s

      LEFT JOIN serie\_language sl ON sl.serie\_id = s.id

      LEFT JOIN download\_serie ds ON ds.serie\_language\_id = sl.id

      LEFT JOIN reproduction\_serie rs ON rs.serie\_language\_id = sl.id

      HAVING COUNT(DISTINCT ds.id) > 0 OR COUNT(DISTINCT rs.id) > 0

      GROUP BY s.title

    UNION

    SELECT 'Documental' Tipo, d.title Titulo,

      COUNT(DISTINCT dd.id) Descargas,

      COUNT(DISTINCT rd.id) Reproducciones,

      MAX(rd.begin\_date) last\_reproduction,

      MAX(dd.begin\_date) last\_download,

      COUNT(DISTINCT dd.id) + COUNT(DISTINCT rd.id) Total

      FROM documentary d

      LEFT JOIN documentary\_language dl ON dl.documentary\_id = d.id

      LEFT JOIN download\_documentary dd ON dd.documentary\_language\_id = dl.id

      LEFT JOIN reproduction\_documentary rd ON rd.documentary\_language\_id = dl.id

      GROUP BY d.title

  )

  ORDER BY Total DESC

) r

WHERE ROWNUM <= 10;

Resultado:

A screen shot of a computer

Description automatically generated

Reporte #10

SELECT g.name Genero,

COUNT(DISTINCT m.id) Peliculas,

COUNT(DISTINCT s.id) Series,

EXTRACT(YEAR FROM MIN(m.year)) "PRIMER PELICULA",

EXTRACT(YEAR FROM MIN(s.begin\_date)) "PRIMER SERIE"

FROM genre g

LEFT JOIN movie\_genre mg ON mg.genre\_id = g.id

LEFT JOIN movie m ON m.id = mg.movie\_id

LEFT JOIN serie\_genre sg ON sg.genre\_id = g.id

LEFT JOIN serie s ON s.id = sg.serie\_id

GROUP BY g.name

ORDER BY MIN(m.year) ASC;

Resultado:

