

Tecnicatura Universitaria en Procesamiento y Explotación de Datos

**INFORME**

**Trabajo Práctico Final de Bases de Datos**

Por

**Armú Yamil, Zamora Rodrigo**

08/06/2023

**INTRODUCCIÓN**

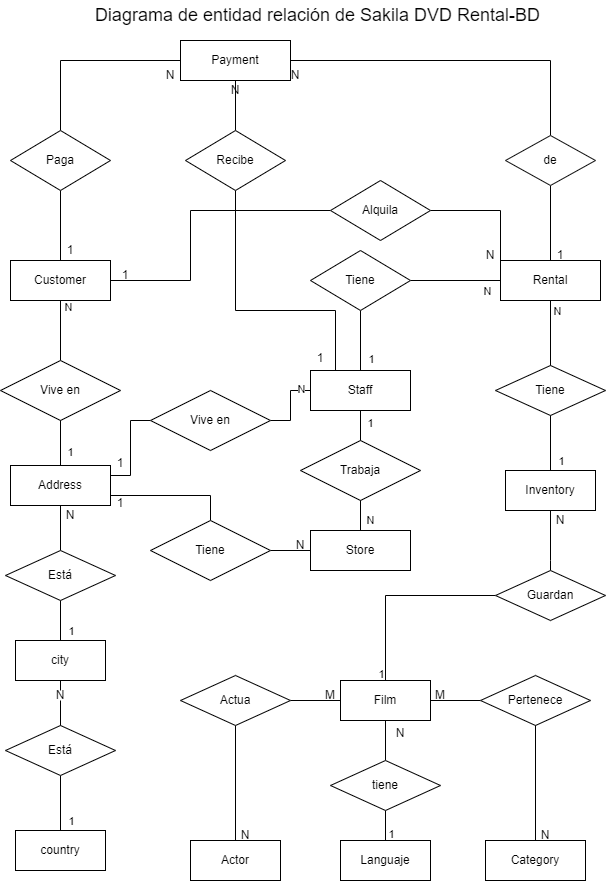
Este trabajo es el resultado de una **investigación** a partir de la **base de datos "Sakila DVD Rental"**. Esta base es utilizada para gestionar un local de alquiler de DVD, y contiene información sobre alquileres, clientes, películas, inventario, entre otros aspectos relevantes.

El objetivo de este trabajo es aplicar las herramientas y conceptos aprendidos en la asignatura **Bases de Datos** de la **Tecnicatura Universitaria en Procesamiento y Explotación de Datos.** Para así explorar la base de datos en cuestión, a fin de comprender su estructura, y definición **DDL** (Lenguaje de Definición de Datos), además elaborar un **DER** (diagrama de entidad-relación) y un diagrama de tablas. Con ello posteriormente poder realizar consultas en lenguaje **SQL,** para extraer datos/información que nos ayuden a responder las preguntas que nos planteamos, por ejemplo: “que películas generaron más dinero” o “en qué país hay más clientes que consuman los DVD”. En última instancia se desarrolla una **interfaz** conectando la base de datos a un script en lenguaje **R**, para realizar un análisis estadístico y representaciones gráficas de la información obtenida.

El desarrollo es abordado en el siguiente orden:

1. DER.
2. Diagrama de tablas.
3. DDL.
4. Planteamiento de tres consultas SQL complejas que incluyen operaciones avanzadas.
5. Desarrollo de una consulta que involucra tres tablas y su respectivo árbol de ejecución optimizado (tres etapas).
6. Interfaz de acceso a datos en lenguaje R, análisis y representación.

**DER (Diagrama entidad-relación)**



El diagrama de entidad-relación de Sakila-DVD lo hicimos utilizando como referencia el diagrama de tablas creado en pgAdmin. Vale mencionar que la entidad “Rental” en nuestra interpretación es la entidad principal, ya que relaciona tres de las entidades más importantes como “Staff”, “Customer” e “Inventory”, dándole sentido lógico al proceso de generación de datos, dónde los clientes alquilan películas de un catálogo disponible en un videoclub que funciona debido al staff o empleados.

Por otra parte, las cardinalidades entre las tablas "film-actor" y "film-category" son de "m" a "n", debido a la existencia de una tabla intermedia que contiene dos llaves primarias que, a su vez, hacen referencia a otras llaves primarias.

Se puede considerar la idea de combinar la relación "vive en" entre el cliente (customer) y el personal (staff) con la dirección (address), en lugar de tener dos relaciones separadas. Sin embargo, en este caso se asume que el personal y el cliente viven en el mismo lugar. otra idea fue pensar payment como una relación en vez de una entidad, si bien tiene llave primaria esta no esta siendo usada, de todas maneras elegimos hacerla como una entidad porque esa fue la manera que encontramos para conservar sus datos “amount numeric”.

Para mejorar la visualización en este informe se incluyó un DER sin los respectivos atributos a cada entidad, es por ello que se adjunta en un enlace el DER completo con atributos:

[DER Sakila-DVD](https://drive.google.com/file/d/1XkIoE46oBVqkWszlGjg4tbENAN2Mo0qZ/view?usp=sharing)

**Diagrama de tablas**

Para elaborar el diagrama de tablas nos ayudamos de la herramienta de PgAdmin de Postgres, la cual permite generar un diagrama sobre la base de datos elegida. En base a dicho diagrama se creó otro modificando el orden de las tablas, a fin de mejorar la lectura y entendimiento de las llaves primarias y foráneas, además de las cardinalidades de las mismas.

Debido al tamaño del diagrama resulta imposible integrarlo a éste informe, por ende para una mejor visualización y comprensión del mismo, se adjunta a continuación en un enlace:

[Diagrama de tablas Sakila-DVD](https://drive.google.com/file/d/1HfFfATFZVagAf-UQNUbW9GCw6FpAX4Yq/view?usp=sharing)

**DDL**

Nuevamente mediante las herramientas de PgAdmin se obtuvo el DDL (Data Definition Lenguage) o Lenguaje de Definición de Datos, correspondiente a la base de datos elegida “Sakila-DVD”. El código que define la base de datos es muy extenso, ya que dicha base se compone de 15 tablas, por ello se adjunta un enlace por fuera del informe.

[DDL Sakila-DVD](https://drive.google.com/file/d/1FAdSs7dmNmprlVJD7rlnssJvdKbEotEt/view?usp=sharing)

**Planteamiento de tres consultas SQL complejas que incluyen operaciones avanzadas.**

A continuación, a fin de obtener información que ayude a responder algunas preguntas que nos hicimos, se realizan tres consultas en SQL en las que se utilizan operaciones avanzadas como JOIN, EXIST, IN, entre otras. Además se enseña la salida de PgAdmin a las mismas.

**-- ¿En qué país hay más clientes?**

SELECT co.country, COUNT(\*) AS total\_clientes

FROM customer AS cl

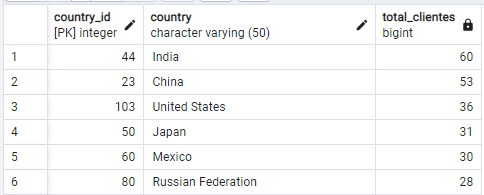
JOIN address AS a ON cl.address\_id = a.address\_id

JOIN city AS ct ON a.city\_id = ct.city\_id

JOIN country AS co ON ct.country\_id = co.country\_id

GROUP BY co.country

ORDER BY total\_clientes DESC;



**--top 3 películas que más dinero generaron**

SELECT f.title, SUM(p.amount) AS total\_amount

FROM payment AS p

JOIN rental AS r ON p.rental\_id = r.rental\_id

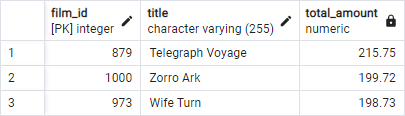
JOIN inventory AS i ON r.inventory\_id = i.inventory\_id

JOIN film AS f ON i.film\_id = f.film\_id

GROUP BY f.title

ORDER BY total\_amount DESC

LIMIT 3;



**--10 peliculas que esten en ingles y sean de acción**

SELECT f.title

FROM film AS f

JOIN language AS l ON f.language\_id = l.language\_id

JOIN film\_category AS fc ON f.film\_id = fc.film\_id

JOIN category AS c ON fc.category\_id = c.category\_id

WHERE l.name = 'English' AND c.name = 'Action'

LIMIT 10;

**Desarrollo de una consulta y su respectivo árbol de ejecución optimizado (tres etapas).**

Se crea una consulta que utiliza tres tablas, contiene una condición de igualdad y una condición de rango con el fin de crear el árbol de ejecución de la consulta en álgebra relacional y luego optimizarlo.

**-- Mostrar el nombre de las películas de terror que duren entre 1 y 2 horas.**

SELECT f.title

FROM film AS f

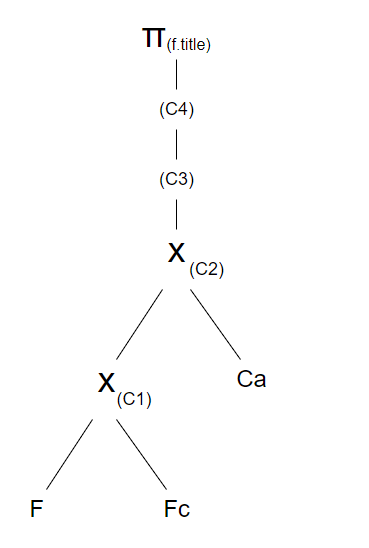
JOIN film\_category AS fc ON fc.film\_id = f.film\_id ⇦C1

JOIN category AS ca ON ca.category\_id = fc.category\_id ⇦C2

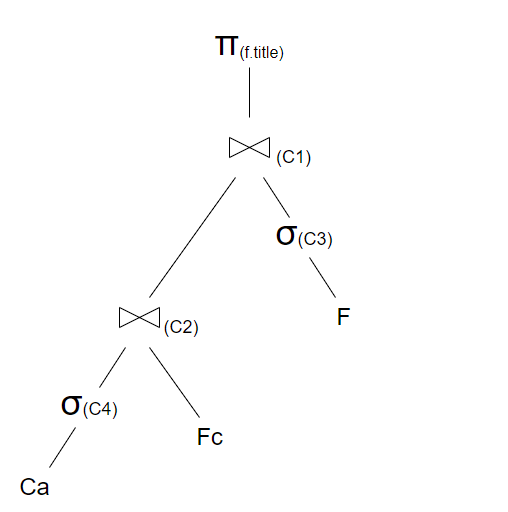
WHERE f.length BETWEEN 60 AND 120 ⇦C3

AND ca.name = 'Horror' ⇦C4

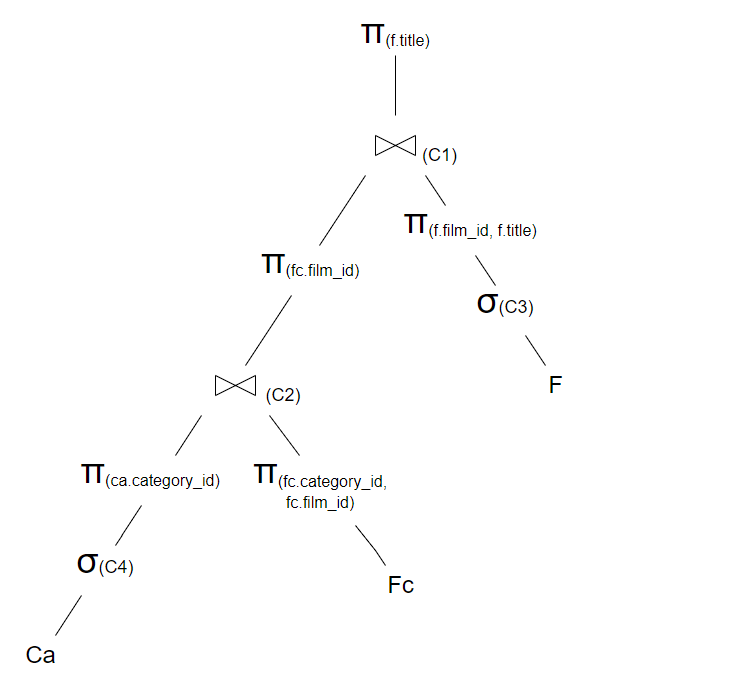
Árbol de ejecución inicial: El árbol de ejecución inicial representa la consulta sin ninguna optimización. Se basa en la estructura original de la consulta y los operadores necesarios para obtener los resultados deseados.



Optimización con σ (Sigma) - Selección: En el segundo árbol de ejecución, se agrega la operación de selección (σ) para filtrar las filas que cumplan ciertas condiciones. En este caso, las condiciones son C4 (ca.name = 'Horror') y C3 (f.length BETWEEN 60 AND 120), además de C1 (fc.film\_id = f.film\_id) y C2 (ca.category\_id = fc.category\_id) condiciones respectivas a las operaciones de JOIN. Al aplicar la operación de selección, se reducen las filas de las tablas involucradas en un subconjunto de datos más específico.



Optimización con Π (Pi) - Proyección: En el tercer árbol de ejecución, se aplica la operación de proyección (Π) para seleccionar columnas específicas de las tablas. Esto implica eliminar las columnas que no serán utilizadas en el proceso y seleccionar únicamente las necesarias para los resultados finales. En este caso, las columnas seleccionadas son “ca.category\_id”, “fc.category\_id”, “fc.film\_id”, “f.film\_id” y “f.title”.



Consulta optimizada:

SELECT SC4.title

FROM (SELECT film\_id, title FROM film WHERE length BETWEEN 60 AND 120) SC4,

(SELECT SC2.film\_id

FROM (SELECT category\_id FROM category WHERE name = 'Horror') SC1,

(SELECT category\_id, film\_id FROM film\_category) SC2

WHERE SC1.category\_id = SC2.category\_id) SC3

WHERE SC4.film\_id = SC3.film\_id

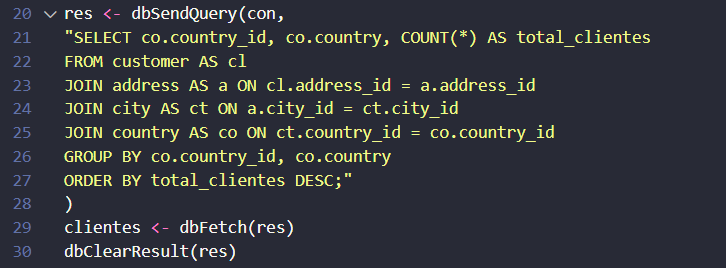
**Interfaz de acceso a datos en lenguaje R, análisis y representación.**

En última instancia se realizó una interfaz de acceso a la base de datos en R, con el fin de crear un gráfico que nos permitiera visualizar de mejor manera los resultados obtenidos mediante las consultas SQL planteadas anteriormente.

Para ello se utilizaron las librerías “DBI” y “RPostgres” para conectar la base de datos al script, por otro lado para el manejo de datos la librería “dplyr”.

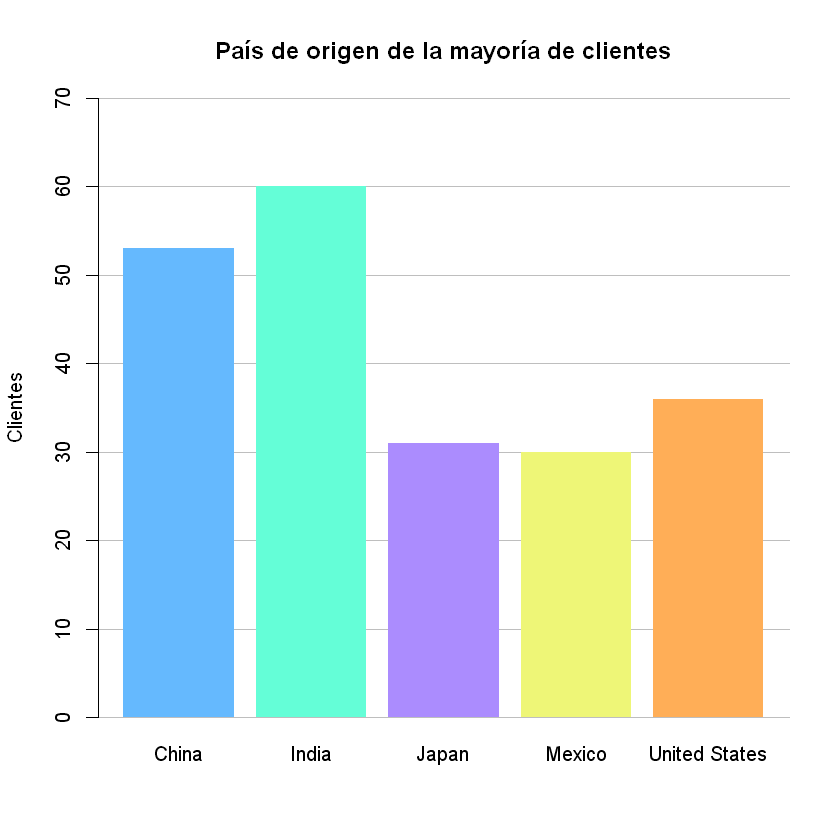


Una vez realizada la conexión se procedió a extraer datos con una consulta SQL.



Con los datos disponibles creamos un gráfico de barras que nos permitió representar la información de una manera clara.

En él podemos observar los países de origen de la mayoría de clientes de la tienda de alquiler de DVD, diferenciados con diferente color y un recuento de la cantidad de clientes.



Script completo: [Interfaz](https://drive.google.com/file/d/1h9mKJ9XjZWJzPzRqq5FE4F7WACJbnk91/view?usp=sharing)

**CONCLUSIÓN**

A modo de conclusión: Pudimos cargar exitosamente la base de datos. Además, hemos sido capaces de formular y ejecutar preguntas complejas utilizando SQL, lo cual pudimos lograr con los conocimientos adquiridos en la materia.

Durante el proceso, descubrimos que generar gráficos del dataset desde una interfaz de acceso a datos, como R en nuestro caso, resultó muy útil para comprender cómo se relaciona esta materia con el resto de nuestra carrera y los conceptos en los que hemos estado trabajando.

Inicialmente, elegimos este dataset con la expectativa de encontrar información sobre películas, como las más vendidas en diferentes países. Sin embargo, al realizar consultas, nos dimos cuenta de que los datos en el dataset son ficticios y no representan información real. Por ejemplo, los nombres de las películas no son reales.

Como trabajos a futuro, nos planteamos cargar datos reales para obtener conclusiones más precisas y generar gráficos más significativos. Esto nos permitirá explorar de manera más efectiva los temas relacionados con películas y extraer información valiosa de la base de datos.