

Disparadores y Vistas en SQL

Samuel Martín Morales

November 7, 2023

Contents

1	Introducción	2
2	Resultados	3
2.1	Ejercicio 1	3
2.2	Ejercicio 2	3
2.2.1	Ventas totales	3
3	Conclusiones	4
4	Bibliografía	5

Chapter 1

Introducción

Para esta cuarta práctica de la asignatura *Administración y Diseño de Bases de Datos* se solicita el empleo de una base de datos que debe de ser restaurada de manera previa a la implementación de una serie de ejercicios que son demandados haciendo uso de dicha base de datos.

En este caso, la base de datos a emplear se denomina como ***alquilerdvd.tar*** y se encuentra disponible en el campus virtual de la asignatura. Pero, puede ser descargada desde el siguiente enlace de GitHub.

Dicha base de datos se encuentra en formato *.tar* por lo que, para poder restaurarla, se debe de emplear el siguiente comando:

```
$ pg_restore -U postgres -d alquilerdvd alquilerdvd.tar
```

Es decir, el comando anterior restaura la base de datos *alquilerdvd* haciendo uso del fichero *alquilerdvd.tar* y empleando el usuario *postgres*.

Una vez restaurada la base de datos, se puede proceder a la realización de los distintos ejercicios.

Chapter 2

Resultados

2.1 Ejercicio 1

Para el primer ejercicio de la práctica, se deben de identificar las distintas tablas, vistas y secuencias que tiene la base de datos que ha sido restaurada.

Tras la carga de la base de datos a partir del fichero con extensión *.tar*, se puede observar que la base de datos *alquilerdvd* cuenta con un total de 15 tablas, 4 vistas y 15 secuencias. Para poder visualizar todos estos datos comentados sobre la base de datos, se hace uso de la terminal interactiva de *PostgreSQL*, es decir, de *psql*, y, una vez dentro de la base de datos se ejecutan los siguientes comandos para obtener los distintos valores obseados anteriormente:

```
# \dt -- Muestra las tablas de la base de datos
# \dv -- Muestra las vistas de la base de datos
# \ds -- Muestra las secuencias de la base de datos
```

2.2 Ejercicio 2

Tras la identificación de las distintas tablas más importantes de la base de datos junto con sus atributos y relaciones entre las distintas tablas, se procede a la implementación de distintas consultas que permitan obtener aquella información que es solicitada en el enunciado del ejercicio.

2.2.1 Ventas totales

Para obtener las ventas totales por categoría de películas ordenadas de manera descendente, se emplea la siguiente consulta:

```
SELECT COUNT(*) AS total_rent, category.name AS category_name
FROM rental
INNER JOIN inventory ON rental.inventory_id = inventory.inventory_id
INNER JOIN film ON inventory.film_id = film.film_id
INNER JOIN film_category ON film.film_id = film_category.film_id
INNER JOIN category ON film_category.category_id = category.category_id
GROUP BY category_name
ORDER BY total_rent DESC;
```

Chapter 3

Conclusiones

Example....

Chapter 4

Bibliografía

Example....