**INSTITUTO TECNOLÓGICO DE COSTA RICA I SEMESTRE 2021**

**INGENIERÍA EN COMPUTACIÓN**

**IC-6600 SISTEMAS OPERATIVOS**

**Tarea Programada 2 – Replicación y OLAP**

Las siguientes son las instrucciones para la documentación.

1. Introducción

* Presentar brevemente el problema en uno o dos párrafos. Puede “reciclar” partes del enunciado de la tarea programada.

1. Ambiente de desarrollo

* Indicar las herramientas usadas para implementar la tarea.

1. Incluir en un archivo adicional los scripts de SQL desarrollados para las diferentes partes del proyecto.
2. Completar la información de la tabla que se incluye al final de este documento.
3. Comentarios finales (estado final del proyecto)  
   Indicar el estado final en que quedó el proyecto, problemas encontrados y limitaciones adicionales.
4. La tarea debe ser entregada presentando un archivo llamado  
    **TP2BD2-ApellidoNombre1- ApellidoNombre2.zip**  
   que contiene un directorio comprimido (ZIP) con la siguiente estructura:  
    *TP2BD2-ApellidoNombre1- ApellidoNombre2*  
    *Archivo\_de\_documentacion*  
    *Scripts de SQL  
    Resultados consultas*  *Archivos\_adicionales*
5. Enviar la tarea y su documentacion por correo electrónico [josee.arayamonge@gmail.com](mailto:josee.arayamonge@gmail.com).

**Usar *asunto*: “TP2BD2-ApellidoNombre1- ApellidoNombre2”**.

**Tabla resumen**

|  |  |  |
| --- | --- | --- |
| **Cargar la base de datos** | **Implement.** | **Comentarios/Observaciones** |
| * toda la base de datos fue cargada exitosamente | **SI** |  |
| **Modificar base de datos** |  |
| * procedimientos almacenados |  |
| insertar nuevo cliente | **SI** |
| registrar un alquiler | **SI** |
| registrar una devolución | **SI** |
| buscar una película | **SI** |
| insertar una nueva película y su inventario | **SI** |
| * roles |  |
| EMP: puede ejecutar 4 primeros procedimientos | **SI/NO** |
| ADMIN: puede ejecutar los procedimientos de EMP más el último | **SI/NO** |
| * usuarios |  |
| emp1: puede hacer login y tiene rol EMP; no tiene ningún otro derecho | **SI/NO** |
| admin1: puede hacer login y tiene rol ADMIN; no tiene ningún otro derecho | **SI/NO** |
| video: dueño de las tablas y procedimientos | **SI/NO** |
| * seguridad |  |
| emp1: puede hacer login y solo tiene los derechos del rol EMP | **SI/NO** |
| admin1: puede hacer login y solo tiene los derechos del rol ADMIN | **SI/NO** |
| video: no puede hacer login; dueño de tablas y procedimientos almacenados | **SI/NO** |
| video: los procedimientos almacenados corren con las credenciales de seguridad de video | **SI/NO** |
| **Replicar base de datos** |  |
| * replicación realizada exitosamente | **SI/NO** |
| **Montar modelo estrella** |  |
| * dimensiones creadas exitosamente | **SI/NO** |
| películas, lugar fecha, lenguaje, duración |  |
| * tabla de hechos creada exitosamente | **SI/NO** |
| * carga de datos exitosa | **SI/NO** |
| dimensiones |  |
| hechos |  |
| * creados los índices de las dimensiones y tablas de hechos | **SI/NO** |
| * procedimientos de consultas |  |
| para un mes dado, sin importar el año, dar para cada categoría de película el número de alquileres realizados | **SI/NO** |
| dar el número de alquileres y el monto cobrado, por duración del préstamo | **SI/NO** |
| hacer un rollup por año y mes para el monto cobrado por alquileres | **SI/NO** |
| hacer un cubo por año y categoría de película para el número de alquileres y el monto cobrado | **SI/NO** |

Consultas usando modelo relacional original

-- para un mes dado, sin importar el año, dar para cada categoría de película el número de alquileres realizados

select date\_part('month',rental\_date) as mes, c.name,

count(\*) as num\_alquileres

from rental r inner join inventory i on r.inventory\_id = i.inventory\_id

inner join film\_category fc on i.film\_id = fc.film\_id

inner join category c on fc.category\_id = c.category\_id

where date\_part('month',rental\_date)=7

group by mes, c.name

order by mes, c.name;

-- dar el número de alquileres y el monto cobrado, por duración del préstamo

select (date\_part ('day', return\_date - rental\_date )) +

ceiling(date\_part ('hour', return\_date - rental\_date ) /24) as duracion,

count(\*), sum(amount)

from rental r inner join payment p on r.rental\_id = p.rental\_id

where return\_date is not null

group by duracion

order by duracion;

-- hacer un rollup por año y mes para el monto cobrado por alquileres

select date\_part('year', rental\_date) as anno,

date\_part('month', rental\_date) as mes,

sum(amount)

from rental r inner join payment p on r.rental\_id = p.rental\_id

group by rollup (anno, mes)

order by anno, mes

-- hacer un cubo por año y categoría de película para el número de alquileres y el monto cobrado

select date\_part('year',rental\_date) as anno, c.name,

count(\*) as num\_alquileres,

sum(p.amount) as monto\_cobrado

--, r.rental\_id, i.inventory\_id, i.film\_id, fc.category\_id, rental\_date

from rental r inner join inventory i on r.inventory\_id = i.inventory\_id

inner join film\_category fc on i.film\_id = fc.film\_id

inner join category c on fc.category\_id = c.category\_id

inner join payment p on r.rental\_id = p.rental\_id

group by cube (anno, c.name)

order by anno, c.name;