

Laboratorio 7 - Vistas y Disparadores

Se cuentan con las siguientes tablas en el servidor del curso:

- **pelicula**(nombre, año, calificacion, votos)
- **actor**(nombre, genero)
- **personaje**(a_nombre, p_nombre, p_año, personaje)
- **resenha**(g_nombre, p_nombre, p_año, opinion, eval)

Las primeras tres tablas están en el esquema **cine** y la última tabla está en el esquema **eval**. En el esquema **eval** usted tendrá privilegios para crear, borrar e insertar y le pedimos que los use con precaución para evitar arruinar el trabajo de los demás. Todas las vistas y otros que se le pidan deben tener un nombre único, para que su trabajo no colisione con el de otros grupos.

Puede consultar el archivo **vistas-disparadores.txt** en U-Cursos para ver ejemplos útiles.

Al final del laboratorio, usted debe entregar un archivo **.txt** con las respuestas de las siguientes preguntas (incluyendo comandos SQL, tiempos, y respuestas breves a las preguntas). Cuando se piden los tiempos, se puede usar **EXPLAIN ANALYZE** y tomar el tiempo de ejecución.

P1. 15 PUNTOS En esta pregunta se desea obtener todas las parejas de películas de la base de datos que han compartido al menos un actor participante (de cualquier género), y la cantidad de actores (distintos) que comparten. El resultado debería contener tuplas de la forma (n_1, a_1, n_2, a_2, k) tal que: n_1 y a_1 sean el nombre y año de la primera película, n_2 y a_2 sean el nombre y año de la segunda película, $k > 0$ sea el número de actores compartido entre ambas películas, y $n_1 < n_2$ o $(n_1 = n_2 \text{ y } a_1 < a_2)$. La última condición evita parejas reflexivas como (n, a, n, a, k) , y parejas simétricas como (n_i, a_i, n_j, a_j, k) si ya tenemos (n_j, a_j, n_i, a_i, k) .

- Escriba una consulta en SQL que obtenga lo pedido; obtenga el tiempo de ejecución y el número de resultados.
- En el esquema **eval** cree una vista que sirva para obtener lo pedido. La vista debe tener un nombre único en el esquema. La vista no puede tener dos columnas con el mismo nombre; se puede usar **AS** en la consulta para cambiar los nombres de las columnas. Luego ejecute la consulta:

```
SELECT * FROM VISTA
```

sobre ella, obteniendo el tiempo de ejecución y el número de resultados.

- Cree una vista materializada que obtenga lo pedido y ejecute la consulta

```
SELECT * FROM VISTA_MATERIALIZADA
```

sobre ella, obteniendo el tiempo de ejecución y el número de resultados.

- P2.** 15 PUNTOS En esta pregunta, se desea obtener, para la película Brazil (1985), todas las *otras* películas con la cual comparte al menos 3 actores. Los resultados deberían contener tuplas de la forma (n, a, k) tal que n sea el nombre de la película, a sea su año, y $k \geq 3$ sea el número de actores (distintos) en esta película y en Brazil (1985).
- (a) Escriba una consulta en SQL que obtenga la respuesta directamente de las tablas bases y registre el tiempo de ejecución y el número de resultados.
 - (b) Escriba una consulta que obtenga los mismos resultados que la **P2.a)** pero esta vez usando (solo) la vista creada en la **P1.b)**; registre el tiempo de ejecución y el número de resultados.
 - (c) Escriba una consulta que obtenga los mismos resultados que la **P2.a)** pero esta vez usando (solo) la vista materializada creada en la **P1.c)**; registre el tiempo de ejecución y el número de resultados. ¿Es este tiempo mejor que el obtenido con las otras alternativas? Cree los índices que crea necesarios sobre la vista materializada para mejorar el tiempo de ejecución y registre el tiempo de ejecución con los índices nuevos.
- P3.** 30 PUNTOS En esta pregunta, usted debe agregar evaluaciones en la tabla `eval.resenha`, donde los atributos `p_nombre` y `p_año` indican la llave de la película a evaluar, `g_nombre` es el nombre del grupo que hace la evaluación y `eval` la calificación (entre 0 y 10). Se le pide que:
- (a) Agregue dos evaluaciones a dos películas distintas de su elección.
 - (b) Cree una vista virtual de todas las evaluaciones que se han hecho en las mismas películas que evaluó usted usando el nombre de su grupo (incluyendo las evaluaciones de su grupo). Debe proyectar todos los atributos de la tabla `eval`. Cree la vista materializada equivalente.
 - (c) Agregue dos evaluaciones nuevas en otras dos películas.
 - (d) Consulte por las evaluaciones tanto en su vista virtual como en la materializada ¿Son iguales? ¿Por qué?
 - (e) Refresque la vista materializada y vuelva a ejecutar las consultas ¿Son iguales? ¿Por qué?
 - (f) Cree un disparador y su correspondiente *stored procedure* para refrescar su vista materializada *solo* cuando haya una actualización o inserción *relevante* para la vista en la tabla `eval.resenha`, es decir, una actualización o inserción que involucre las evaluaciones de su grupo o las evaluaciones de las mismas películas hechas por otros grupos (aquí, el archivo `vistas-disparadores.txt` en UCursos puede ser muy útil). Agregue una evaluación más para su grupo y verifique que la vista materializada se actualiza automáticamente y correctamente.