

Informe Grupo 03

Proyecto Especial de Cátedra:

“Sistema de publicidad de Juegos en línea”

Integrantes: Martin Aguirre, Nicolas Erasun, Tomas Porcaro.

Materia: Bases de Datos.

Profesores: Viviana Ferraggine, Sebastian Villar.

Año: 2020.

Carrera: TUDAI.

Introducción:

Recibimos un pedido de gestión de base de datos, de la empresa “**PJL Sistemas Tandil**” donde se requería realizar un conjunto de controles y servicios sobre los esquemas dados por la misma.

La base de datos a gestionar mantiene un sistema de publicidad de juegos en línea que están organizados en categorías, las cuales pueden ser agregadas, removidas o modificadas y poseen un juego o un nivel determinado asociado a cada una. Estos juegos asociados pueden tener distintos niveles de dificultad designados según su contenido. Los juegos son jugados por varios usuarios y tienen el privilegio de votar y recomendar estos, a su vez también pueden comentarlos.

Se nos pidió realizar distintas restricciones:

- *La fecha del primer comentario tiene que ser anterior a la fecha del último comentario si este no es nulo.*
- *Cada usuario sólo puede comentar una vez al día cada juego.*
- *Un usuario no puede recomendar un juego si no ha votado previamente dicho juego.*
- *Un usuario no puede comentar un juego que no ha jugado.*

Otro requerimiento fue la implementación de los siguientes servicios:

- *Se debe mantener sincronizadas las tablas COMENTA y COMENTARIO*
- *Dado un patrón de búsqueda devolver todos los datos de el o los usuarios junto con la cantidad de juegos que ha jugado y la cantidad de votos que ha realizado*

También se requirieron la implementación de 4 vistas:

- *Listar todos los comentarios realizados durante el último mes descartando aquellos juegos de la Categoría “Sin Categorías”.*
- *Listar aquellos usuarios que han comentado **TODOS** los juegos durante el último año, teniendo en cuenta que sólo pueden comentar aquellos juegos que han jugado.*
- *Realizar el ranking de los 20 juegos mejor puntuados por los Usuarios. El ranking debe ser generado considerando el promedio del valor puntuado por los usuarios y que el juego haya sido calificado más de 5 veces.*
- *Generar una vista con los 10 juegos más jugados.*

Desarrollo:

Restricciones:

Se pudo controlar que la fecha del primer comentario sea anterior a la del último comentario (o este sea nulo) mediante la implementación de una restricción de tupla sobre la tabla **GR03_comenta**, se eligió este tipo de restricción de integridad ya que el DBMS soporta este tipo de operaciones.

Para los otros tres requisitos se implementaron restricciones de integridad mediante el uso combinado de funciones y triggers. El requisito de un comentario diario por usuario se resolvió con una restricción de tipo tabla sobre **GR03_comentario**, controlando que el usuario no posea una tupla con la fecha actual (al momento de insertar), se verificó el correcto funcionamiento de esta restricción mediante la inserción de la tupla (1,9,'2020-01-11','2020-01-08') generando el mensaje de violación de constraint .

Mientras que las restricciones al usuario que le impidiesen recomendar juegos que no haya votado y comentar juegos que no haya jugado, se pudieron solventar mediante el uso de dos aserciones (de tipo base de datos o general), como las aserciones y las restricciones de tipo tabla no son soportadas en PostgreSQL se utilizó la combinación de trigger/función para resolver estas restricciones, en el primer caso se controló que el exista en la tabla **GR03_voto** una tupla que corresponda al usuario y al juego que será insertado en la tabla **GR03_recomendacion**, y en el segundo caso se controló que exista en la tabla **GR03_juega** una tupla que corresponda al usuario y al juego que será insertado en la tabla **GR03_comentario**. Al insertar los datos que violan estas restricciones se obtiene el mensaje que cargamos dentro de la sentencia RAISE EXCEPTION en la función del trigger.

Servicios:

Luego se implementaron los servicios, el primero se resolvió mediante un trigger AFTER en la tabla **GR03_comentario**, está siendo disparado cada vez que se insertaba algo, se actualizaba en id de juego, usuario o la fecha del comentario y también cuando se borraba la última tupla insertada en comentario. Cada vez que se insertaba algo en la tabla **GR03_comentario**, este trigger debiera si es la primera vez que un usuario comentaba un mismo juego insertar el id del usuario con el id del juego y la fecha del primer comentario con la fecha correspondiente, a su vez poniendo la fecha del último comentario en nulo; en cambio cuando un usuario ya comentó alguna vez ese juego debiera actualizar la tabla comenta con sus respectivos datos dejando la fecha del primer comentario como esta y insertando la fecha del último comentario correctamente.

El siguiente servicio se resolvió mediante una función que tiene como retorno una tabla y como parámetro un varchar (string). esta función ensambla las tablas **GR03_usuario**, **GR03_juega**, **GR03_voto** para extraer los datos del usuario más la cantidad de juegos que ha jugado y la cantidad de votos que ha realizado que siempre y cuando contengan en su apellido el parámetro dado.

Vistas:

Se generaron cuatro vistas para el proyecto, para listar los comentarios realizados el último mes excluyendo los pertenecientes a juegos sin categorizar se generó una vista actualizable de la tabla **GR03_comentario** de acuerdo a los requisitos de PostgreSQL.

Para el listado de los usuarios que han comentado todos los juegos durante el último año se creó una vista actualizable de la tabla **GR03_usuario**, donde el requisito de solo comentar los juegos que se hayan jugado ya fue controlado en el trigger **TR_GR03_COMENTAR_JUEGO**.

Los 20 juegos más puntuados se pudieron obtener mediante la creación de una vista NO actualizable ya que la misma incluye el ensamble de dos tablas mediante una sentencia JOIN.

Finalmente los 10 juegos más jugados se pudieron obtener mediante la creación de dos vistas, siendo estas alternativas entre sí, una vista actualizable que devuelve los 10 juegos más jugados pero sin un orden determinado o mediante la creación de una vista NO actualizable que permite listar los juegos 10 juegos, ordenados por cantidad de veces que se jugaron cada uno (ver alternativa comentada).

Conclusión:

Mediante el desarrollo de este Proyecto de Cátedra se evidenció y se plasmaron los conocimientos dados a través de este cuatrimestre de Bases de Datos, utilizamos todos los conocimientos aprendidos a lo largo de estos 4 meses tales como la implementación de restricciones de integridad (en este caso de tupla y de base de datos o general), la implementación de triggers y funciones, la implementación de vistas, etc.