
INTRODUCCIÓN

El Trabajo Práctico Especial consiste en la resolución de un conjunto de controles y servicios sobre una base de datos que mantiene un **Sistema de Publicidad de Juegos en Línea** la empresa “**PJL Sistemas Tandil**”. Las consignas para su desarrollo se indican en la sección “**Pautas de Desarrollo**” y el **esquema de la base de datos** se encuentra disponible en formato SQL exportado desde Vertabelo.

DEFINICIÓN DEL UNIVERSO DEL DISCURSO

La aplicación sirve de soporte a un sistema de juegos gratis en línea. Para poder realizar la publicidad de un juego, los usuarios deben estar registrados en el sitio y hay diferentes tipos de usuarios (competidores, socializados, etc.)

Los juegos se organizan en categorías, por ejemplo (juegos de acción, autos, aventuras, didácticos, etc.) las que pueden variar con el tiempo, agregándose nuevas o quitando algunas. Las categorías tienen asociado además un nivel de juego (inicial, intermedio, avanzado, etc).

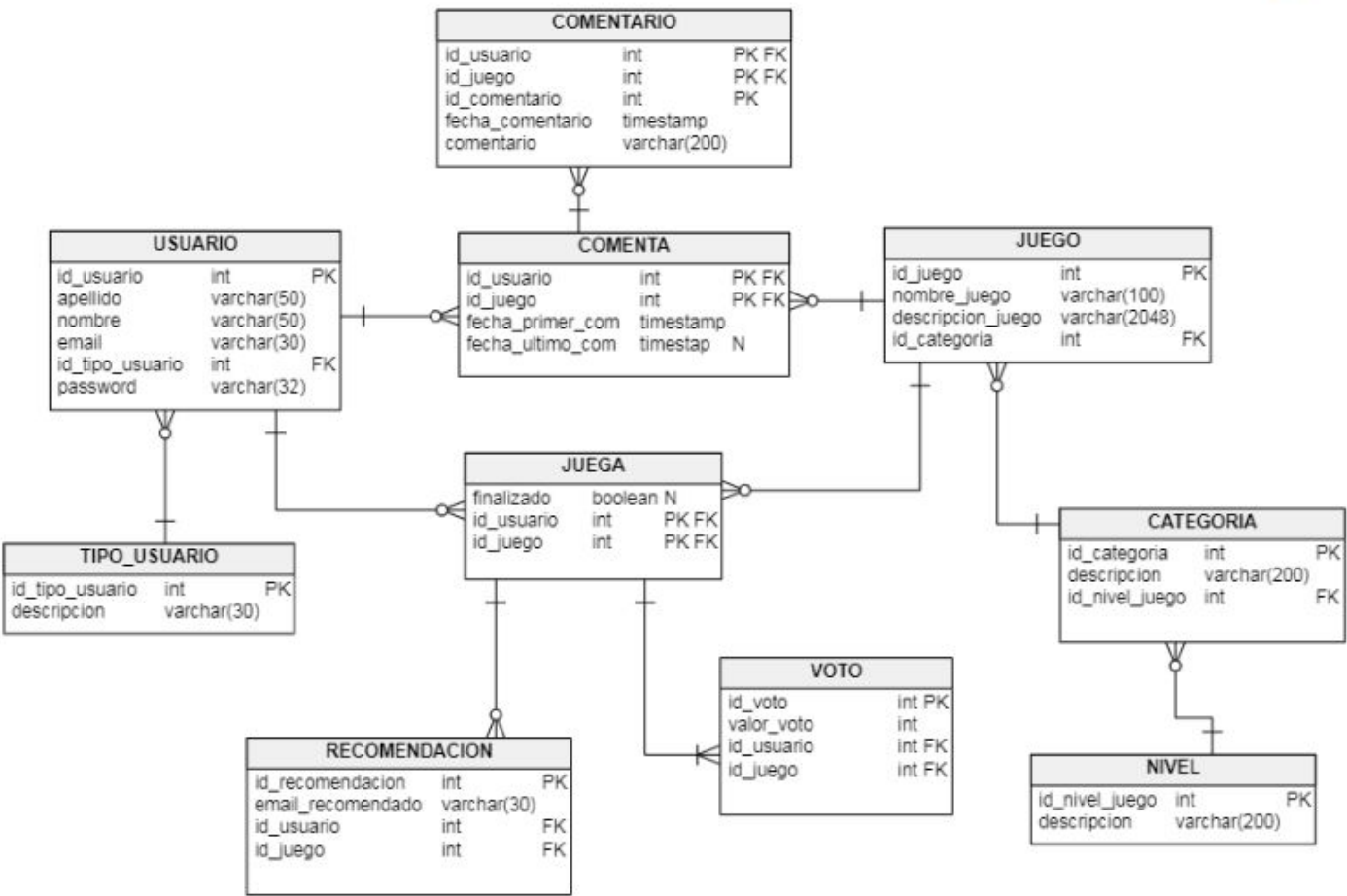
Los juegos son jugados por varios usuarios, los cuales pueden recomendar estos juegos a otros a fin de permitir incrementar la popularidad. En cada uno de los juegos es permitido realizar comentarios con el fin de intercambiar información entre usuarios, luego estos comentarios son presentados cronológicamente junto a la información del juego.

También se permite votar en cada uno de los juegos para incrementar su popularidad y no puede ser modificado ni borrado una vez efectuado. Solo los usuarios que han jugado tendrán la opción de votar y recomendar un juego en particular.

Los juegos de determinadas categorías pueden tener diferentes niveles: principiante, intermedio, avanzado, experto (o algunos de estos). Por razones estratégicas, definidas por los patrocinadores del sitio, se rotulan los comentarios publicados por usuarios que están en diferentes niveles, para un determinado juego, de manera que puedan ser recuperados rápidamente.

El sitio a tomar como ejemplo es <http://www.adondejuegos.com>, siempre teniendo en cuenta que la funcionalidad requerida se limita a la especificada en este enunciado.

DIAGRAMA DEL ESQUEMA DE LA BASE DE DATOS



TAREAS A DESARROLLAR

A partir del anterior esquema original de la BD se requiere:

A. AJUSTE DEL ESQUEMA

- Debe crear su esquema a partir del archivo SQL de Vertabelo provisto por la cátedra.
- Debe ajustar dicho esquema según las **pautas de nomenclatura** indicada en la sección *Reglas de Nombres*.
- Debe generar un script de generación del esquema denominado **GXX_Creación.sql** que cree todas las tablas con sus restricciones de nulidad, primary key y foreign key con las tablas del diagrama de la cátedra y las que cada grupo agregó para las modificaciones.

B. ELABORACIÓN DE RESTRICCIONES

Para cada una de las restricciones/reglas del negocio enunciadas a continuación:

1. Escriba la restricción de la manera que considere más apropiada en **SQL estándar declarativo**, indicando su tipo y características correspondientes. NO se tendrán en cuenta las tipificaciones correctas pero con implementaciones inadecuadas.(incluya el desarrollo en el informe)
2. Impleméntela en **PostgreSQL de la forma más adecuada**, según las posibilidades que ofrece el DBMS (incorpore la solución **al script GXX_Cambios.sql**)
3. Provea una sentencia de modificación, borrado y/o inserción según corresponda, de modo que promueva la activación de la restricción (incorpórela comentada al script **GXX_Cambios.sql** de modo que la ejecución del mismo no de error). Explique brevemente la respuesta por parte del DBMS (en el informe).

Se deben realizar los siguientes controles en el esquema de datos:

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

C. SERVICIOS

Implemente lo siguiente con el recurso que crea más conveniente (triggers, procedimientos y/o vistas). Si necesita para este ítem crear tablas adicionales al esquema creado en el script de creación (GXX_Creacion.sql) incorpórelas en el script **GXX_Cambios.sql** .

1- Se debe mantener sincronizadas las tablas COMENTA y COMENTARIO en los siguientes aspectos:

- a. La primera vez que se inserta un comentario de un usuario para un juego se debe hacer el insert conjunto en ambas tablas, colocando la fecha del primer comentario y última fecha comentario en nulo.
- b. Los posteriores comentarios sólo deben modificar la fecha de último comentario e insertar en COMENTARIO

2- 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.

D. DEFINICIÓN DE VISTAS

Escriba la sentencia SQL para crear las vistas detalladas a continuación e incorpórelas en el script **GXX_Cambios.sql** . En cada caso comente y ejemplifique (en el informe) si es actualizable o no, indicando la/s causas/s. Vista:

1. *COMENTARIOS_MES: Listar todos los comentarios realizados durante el último mes descartando aquellos juegos de la Categoría "Sin Categorías".*
2. *USUARIOS_COMENTADORES: 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.*
3. *LOS_20_JUEGOS_MAS_PUNTUADOS: 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 hubiera sido calificado más de 5 veces.*
4. *LOS_10_JUEGOS_MAS_JUGADOS: Generar una vista con los 10 juegos más jugados.*

PAUTAS DE ENTREGA

Consideraciones Importantes

- Se recomienda leer el enunciado completo del trabajo antes de resolverlo
- Se deberán respetar
 - las pautas de entrega indicadas en cada parte del enunciado y
 - las pautas de nomenclatura para todos los objetos del esquema de la base de datos, indicadas en la sección Reglas de Nombres
- Los scripts entregados se probarán y testearán en **PostgreSQL V.11 utilizando DataGrip**. Cada grupo debe asegurarse que los scripts ejecuten completos sin errores para que la cátedra pueda proceder a la corrección del TPE, en el orden establecido; por esto es recomendable que los chequee debidamente a fin de asegurar que los mismos se ejecuten sin inconvenientes.
- Todo el **código SQL entregado debe estar optimizado**, es decir utilizando estrictamente los atributos, tablas y/u operaciones necesarias, de acuerdo a lo requerido.

Nota: Se descontarán hasta 2 puntos de la nota final por no respetar las pautas de entrega, reglas de nombres solicitadas y/o que se deban re-entregar los script por tener errores al momento de testearlos.

Entrega del TPE

Se deberá completar una tarea (ENTREGA) en la plataforma Moodle en la que se deberán subir los **archivos scripts solicitados** en forma separada. En la misma tarea se confirmará el día siguiente a la entrega, si los scripts corrieron sin errores o, de haber advertido algún inconveniente en su ejecución, se informará el primer error encontrado.

Nota: sólo un integrante del grupo debe hacer la entrega, no es necesario que la realicen los 3.

Se deberá entregar un informe en pdf (de **no más de 5 páginas**), explicando concisamente el trabajo desarrollado y 3 (tres) scripts, que se deben **poder ejecutar en orden (detallados en los ítems 1, 2 y 3) y sin producir errores**:

1. script **GXX_Creación.sql**, que deberá contener la creación de lo solicitado en el punto **A**. Nota: cambiar los nombres de los objetos para ajustarlos a los de cada grupo (GXX_).
2. script **GXX_Cambios.sql**, que deberá incluir la implementación de las restricciones, las vistas y los servicios detallados en los puntos **B, C y D**. Nota: Incluir como comentario dentro del script el ítem que resuelve en cada caso.

3. script **GXX_Borrado.sql**, que incluya las sentencias de borrado de todos los objetos del esquema (tanto los creados por el script **GXX_Creacion.sql** como los objetos incorporados mediante el script **GXX_Cambios.sql**)
4. Informe **GXX_Informe.pdf**, en el que se expliquen y justifiquen las decisiones asociadas a la implementación de cada una de las restricciones de integridad, tanto en forma declarativa según el estándar SQL como de forma procedural (para aquellas que declarativamente no soporta PostgreSQL), la definición de las vistas y de los servicios requeridos.

Orden de la Ejecución de Prueba

El orden de la ejecución de los scripts de la prueba que realizará la cátedra será el siguiente:

1. GXX_Creacion.sql
2. GXX_Cambios.sql
3. GXX_Datos.sql
4. GXX_Borrado.sql

REGLAS DE NOMBRES

Para el desarrollo y corrección de los esquemas se seguirán estos estándares que permiten identificar cada trabajo y objeto con nombres completos y descriptivos.

Consideraciones Generales

El prefijo de grupo **<GRXX>** se usará en tablas, vistas, procedimientos almacenados, triggers, índices y constraints. XX se deberá reemplazar por el número de grupo asignado por la cátedra en cada caso. La nomenclatura a seguir es:

1. Tablas/Vistas: **<GRXX>_<nombre_tabla o vista>**
2. Índices: **IDX_<GRXX>_<nombre_indice>**
3. Constraints

- **Clave Primaria: PK_<GRXX>_<nnnn>**

donde: nnnn = el nombre de la tabla para la que la constraint se construye (Por ejemplo: PK_GRXX_Beneficiario)

- **Clave Extranjera: FK_<GRXX>_<pppp>_<cccc>**

donde pppp = el nombre de la tabla referenciante u original, cccc = el nombre de la tabla referenciada.

- **Unique: UQ_<GRXX>_<nnnn>_<cccc>**

donde: nnnn = el nombre de la tabla para la que la constraint se construye, cccc = el nombre del campo interviniente.

4. Procedimientos, Triggers y Funciones

- **Triggers: TR_<GRXX>_<tbl>_<nnnn>**

siendo <nnnn> el nombre que considere adecuado para el trigger y <tbl> la tabla a la cual está asociada el trigger.

- **Procedimientos: PR_<GRXX>_<nnnn>**

siendo <nnnn> el nombre que considere adecuado para el procedimiento.

- **Funciones: FN_<GRXX>_<nnnn>**

siendo <nnnn> el nombre que considere adecuado para la función.

- **Funciones de triggers: TRFN_<GRXX>_<nnnn>**

siendo <nnnn> el nombre que considere adecuado para la función de trigger.



Recordar al momento de realizar los script que:

Aquellos insert, update o delete que provocan fallos de por alguna restricción, es decir los que son de test de restricciones deben colocarse como comentarios /* */.

Recuerde colocarle CASCADE a las sentencias drop table si realiza el borrado de las tablas en cualquier orden, sino tiene que respetar el orden dado por las foreign key.

Una función que es llamada por un trigger debe estar dentro del script siempre antes del create trigger que la invoca.

Recuerde que aquellas sentencias que Postgresql no soporta deben ir comentariadas /* */