Universidad ORT Uruguay

Facultad de Ingeniería

Martin Alonso y Juan Carriquiry

Bases de datos 2

Segundo obligatorio doc parte 2

Grupo M5A

Docente: Cecilia Belleti

Docente: Tatiana Bellon

Formulario de Antecedentes
Curso Materia

Nro. Estudiante: **291799**
Nombres: **Martin**
Apellidos: **Alonso**
Grupo/Turno: **5A/Matutino**

Nro. Estudiante: **310190**
Nombres: **Juan**
Apellidos: **Carriquiry**
Grupo/Turno: **5A/Matutino**

1. Tabla de contenido

1) Supuestos:

a) Supuesto 1:

Usamos las notas de la parte anterior.

b) Supuesto 2:

No pondremos foto del código ya que es relativamente largo y está el archivo presente en la entrega, pero si explicaremos el flujo de trabajo que tiene:

c) Supuesto 3:

Asumimos implementada la parte anterior

d) Supuesto 4:

En el primer requerimiento asumimos que la misión ya está completada.

2) Primer requerimiento

Supuesto:
Asumimos que la misión ya está completada.

a) Validaciones:

Primero, consultamos la tabla Personaje_Mision para verificar que exista un registro que vincule al personaje con la misión.

Luego, verificamos que el estado de esa misión para ese personaje sea 'Completada'.

Finalmente, nos aseguramos que el campo recompensas_recibidas esté en 'N', asegurando que los premios no hayan sido entregados previamente.

Si alguna de estas validaciones falla, la ejecución se detiene y se lanza un error específico (RAISE_APPLICATION_ERROR) para informar la causa del fallo.

b) Procesamiento y entrega de las recompensas

Monedas y Experiencia: Se obtienen los montos de monedas y experiencia de la tabla Recompensa y se actualizan los acumuladores en la tabla Personaje.

Ítems: Se utiliza un Cursor para recorrer todos los ítems asociados a la recompensa en la tabla Recompensa_Posee_Item. Por cada ítem encontrado, se realiza un INSERT en la tabla Personaje_Posee_Items, añadiéndolo al inventario del personaje.

c) Actualización del estado de la recompensa:

Una vez que todas las recompensas han sido asignadas sin errores, se actualiza la información en la tabla Personaje_Mision, cambiando recompensas_recibidas a 'S'.

d) Manejo de errores:

Si todos los pasos anteriores se completan con éxito, se ejecuta un COMMIT para hacer permanentes todos los cambios en la base de datos.

El procedimiento incluye un bloque EXCEPTION general. Si cualquier error inesperado ocurre durante el proceso, se ejecuta un ROLLBACK para deshacer todas las operaciones realizadas hasta el momento, esto es lo que le aporta la atomicidad a la transacción.

3) Segundo Requerimiento

a) Modificaciones DDL

Aquí es importante detallar que modificamos el DDL para que Ítem ya no se más una herencia y pase a ser una tabla normal la cual para gestionar el tipo del item tiene un atributo “categoría”La posibilidad de filtrar por una categoría de ítem de manera opcional se logró definiendo el parámetro de entrada p_categoria con un valor por defecto NULL.

La consulta principal del procedimiento contiene una cláusula WHERE que maneja esta opcionalidad: WHERE (p_categoria IS NULL OR i.categoria = p_categoria). De esta forma:

Si el procedimiento se llama sin el parámetro, p_categoria es NULL, la condición se cumple para todas las filas y el reporte es general.

Si se provee una categoría, la condición filtra los resultados para que coincidan con la categoría especificada.

Flujo de trabajo:

b) Lista de top items

Une las tablas Personaje_Posee_Items e Items para acceder tanto a la información de equipamiento como a los detalles de cada ítem (incluida su nueva columna categoría).

Filtra los registros para considerar únicamente los ítems que están equipados (equipado = 'S').

Utiliza COUNT y GROUP BY para contar las ocurrencias de cada ítem.

Ordena los resultados de forma descendente y utiliza FETCH FIRST 10 ROWS ONLY para obtener el Top 10.

c) Visualización del resultado:

El resultado de la consulta, que puede contener hasta 10 filas, se maneja a través de un Cursor de PL/SQL. El procedimiento itera sobre este cursor para leer cada fila y la imprime en la consola de forma clara y formateada.

4) Tercer requerimiento

Este con diferencia fue el que más cambios requirió,

a) Cambios en el DDL:

Para poder llevar los registros de cuando un personaje aumenta de nivel se hace la tabla “Log_Aumento_Nivel”.
Para llevar el registro de los usuarios que ya recibieron el premio diario se crea la tabla “Log_Premio_Diario”.

b) Triggers adicionales:

Para poder cargar de datos la tabla de “Log_Aumento_Nivel” se crea el trigger “TRG_LOG_NIVEL” que lo unico que hace es al subir un personaje de nivel crea el registro.

Para poder llevar el registro de cuando se hacen las misiones se implementa el trigger “TRG_ACTUALIZAR_FECHA_MISION” que se encarga de que al cambiar el estado de una mision a “Completada” le asigna al registro en “Personaje_Mision” la

fecha actual del sistema

c) Búsqueda de candidatos:

jugadores_level_up: La primera subconsulta revisa la tabla Log_Aumento_Nivel. Para cada jugador, busca todos sus registros de subida de nivel de las últimas 24 horas y calcula la diferencia entre el nivel más alto y el más bajo en ese período. Si la diferencia es 3 o más, mete al jugador en la lista.

jugadores_mision_completa: La segunda subconsulta revisa Personaje_Mision y la une con Misiones. Busca a todos los jugadores que tengan una misión completada en las últimas 24 horas y cuyo tipo de misión (estado) sea 'Principal'.

INTERSECT cruza las dos listas anteriores y devuelve solo los jugadores que están en ambas, cumpliendo así los dos primeros requisitos.
MINUS toma esa lista final de candidatos y le resta todos los jugadores que ya aparecen en Log_Premio_Diario con la fecha de hoy. Esto garantiza la idempotencia.

d) El Proceso de Premiación (El LOOP)

El procedimiento luego utiliza un LOOP para recorrer cada candidato de la lista generada por el cursor. Para cada uno, realiza las siguientes acciones:

Crea un SAVEPOINT: Antes de empezar a procesar a un jugador, se crea un "punto de guardado". Esto es como un mini-checkpoint dentro de la gran transacción. Si algo falla con este jugador, podemos volver a este punto sin afectar al resto.

Para elegir la recompensa aleatoria .Primero, genera un número aleatorio entre 0 y 1 Usa IF/ELSIF para asignar una rareza ('Comun', 'Rara', 'Epica', 'Legendaria') basada en probabilidades (ej. 70% de chance de que sea 'Comun').

Luego, hace una consulta a la tabla Items para seleccionar un único ítem al azar que tenga la rareza elegida.

Registra la entrega del Ítem: Realiza un INSERT en la tabla Personaje_Posee_Items para añadir el nuevo ítem al inventario del personaje.

Registra la Entrega: Inmediatamente después, realiza un INSERT en Log_Premio_Diario para dejar constancia de que este jugador ya recibió su premio hoy.

5) Cuarto requerimiento

a) Validaciones previas:

Verifica que no sea un auto-intercambio: Comprueba que p_email_jugador1 y p_email_jugador2 no sean el mismo.

Verifica que los ítems sean intercambiables: Consulta la tabla Items para asegurarse de que la columna intercambiable de ambos ítems esté marcada como 'S'.

Verifica la Posesión de los Ítems: Consulta la tabla “Personaje_Posee_Items” para confirmar que el jugador 1 realmente posee el ítem 1, y que el jugador 2 posee el ítem 2.

b) Bloqueo del recurso para la integridad transaccional (Select for Update):

Usamos un LOOP vacío (FOR rec IN (...) LOOP NULL; END LOOP;) porque es una forma de ejecutar esta consulta de bloqueo. No nos interesa leer los datos, solo bloquear las filas.

c) Intercambio:

Se ejecuta una sentencia UPDATE para cambiar el dueño del item1, asignándole el emailJugador y el idPersonaje del jugador 2.

Se ejecuta una segunda sentencia UPDATE para cambiar el dueño del item2, asignándole los datos del jugador 1.

d) Manejo de errores y rollBack:

Este es el paso que garantiza la atomicidad ya que si algo sale mal revertimos la transacción.

Pruebas:

a) Nota:

Es importante resaltar que hemos utilizado inteligencia artificial generativa como ChatGPT o Gemini para la generación de datos de prueba y darles un formato más atractivo y entendible.

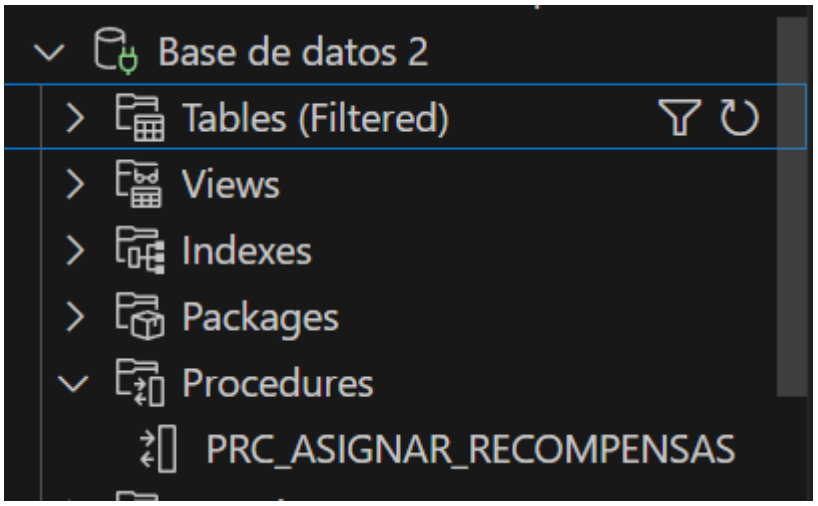
b) Compilación de procedimientos:

Es importante tener los procedimientos compilados previa a la ejecución del archivo de pruebas, compilaremos 1 como demostración:

Buscas el archivo (para esta demo SCRIPTS_NUEVOS\Parte2\Parte2.1.sql) y lo ejecutas, mensaje esperado:

```
Procedure PRC_ASIGNAR_RECOMPENSAS compiled
```

Puedes verificar que se compiló en la extensión de VsCode yendo a la pestaña de procedures:



c) Ejecución:

Ahora podemos ir al archivo en SCRIPTS_NUEVOS\Parte2\Pruebas.sql y ejecutarlo

IMPORTANTE: Para evitar errores del tipo que valores de pruebas pasadas puedan afectar a las futuras se recomienda ejecutar por bloques, osea primero ejecuto la carga de datos de prueba y luego el bloque de las pruebas del servicio 2.1, luego vuelvo a ejecutar la carga de datos y luego las pruebas del servicio 2.2 y así seguir.

1) Primer bloque:

```
--- INICIO PRUEBAS: 2.1 PRC_ASIGNAR_RECOMPENSAS ---


PL/SQL procedure successfully completed.

-- Caso 1.1 (Feliz): Reclamando recompensas para Wukong, misión 101.
   Monedas ANTES: 5000
Recompensas asignadas exitosamente al personaje wukong@monkey.com
   Monedas DESPUÉS: 6000
   Verificando estado: S


PL/SQL procedure successfully completed.

-- Caso 1.2 (Triste): Intentando reclamar de nuevo las recompensas.
Error al asignar recompensas: ORA-20004: Las recompensas para esta misión ya fueron asignadas a este persona
>> ÉXITO: El procedimiento bloqueó la operación con el error esperado.


PL/SQL procedure successfully completed.

-- Caso 1.3 (Triste): Intentando reclamar recompensas de una misión no completada.
Error al asignar recompensas: ORA-20003: La misión aún no ha sido completada por el personaje.
>> ÉXITO: El procedimiento bloqueó la operación con el error esperado.


PL/SQL procedure successfully completed.


Commit complete.
```

2) Segundo bloque:

```
--- INICIO PRUEBAS: 2.2 PRC_TOP_ITEMS_EQUIPADOS ---
-- Caso 2.1 (Feliz): Top 10 general (sin filtro).


PL/SQL procedure successfully completed.

--- Top 10 Items Más Equipados (Todas las Categorías) ---
------------------------------------------------------
Item: Espada de Jade           Categoría: Arma          Equipado: 2 veces
Item: Armadura de Fuego        Categoría: Armadura      Equipado: 1 veces
------------------------------------------------------


PL/SQL procedure successfully completed.

-- Caso 2.2 (Feliz): Top 10 filtrando por categoría 'Arma'.


PL/SQL procedure successfully completed.

--- Top 10 Items Más Equipados (Categoría: Arma) ---
------------------------------------------------------
Item: Espada de Jade           Categoría: Arma          Equipado: 2 veces
------------------------------------------------------


PL/SQL procedure successfully completed.


Commit complete.
```

3) Tercer Bloque:

```
--- INICIO PRUEBAS: 2.3 PRC_ACREDITAR_PREMIO_DIARIO ---
-- Caso 3.1 (Feliz): Ejecutando el premio diario por primera vez.


PL/SQL procedure successfully completed.

Premio (Daga lenta) entregado a wukong@monkey.com
Proceso de acreditación de premios diarios finalizado.


PL/SQL procedure successfully completed.

-- Caso 3.2 (Triste - Idempotencia): Ejecutando el premio por segunda vez. No debería hacer nada.


PL/SQL procedure successfully completed.

Proceso de acreditación de premios diarios finalizado.


PL/SQL procedure successfully completed.


Commit complete.
```

4) Bloque 4:

```
--- INICIO PRUEBAS: 2.4 PRC_INTERCAMBIAR_ITEMS ---


PL/SQL procedure successfully completed.

-- Caso 4.1 (Triste): Wukong intenta intercambiar su Reliquia Celestial (no intercambiable).
Error durante el intercambio. La transacción ha sido revertida. Error: ORA-20011: Uno o ambos ítems no son ir
>> ÉXITO: El procedimiento bloqueó el intercambio de un ítem no permitido.


PL/SQL procedure successfully completed.

-- Caso 4.2 (Feliz): Wukong intercambia su 'Armadura de Fuego' por la 'Daga Veloz' de Tripitaka.
   Items de Wukong ANTES:
   - Armadura de Fuego
   - Daga lenta
   - Espada de Jade
   - Reliquia Celestial
   Items de Tripitaka ANTES:
   - Daga Veloz
   - Espada de Jade
Intercambio realizado exitosamente.
   Items de Wukong DESPUÉS:
   - Daga Veloz
   - Daga lenta
   - Espada de Jade
   - Reliquia Celestial
   Items de Tripitaka DESPUÉS:
   - Armadura de Fuego
   - Espada de Jade


PL/SQL procedure successfully completed.


Commit complete.
```