

1. - DESCRIPCIÓN DE LA OPERATIVA

Sobre la nueva base de datos relacional de la fundación de Bib-buses (FOUNDICU® Org.), es necesario desarrollar algunos elementos operativos. En concreto consultas, vistas, procedimientos y disparadores. La descripción de estos elementos es la siguiente:

1.1.- Consultas (2 puntos, 1 pt c/u)

- a) BoreBooks: libros con ediciones en, al menos, tres idiomas (*language*) distintos, de los que nunca se haya prestado ninguna copia.
- b) Informe de Empleados: para cada conductor, proporcionar su nombre completo, edad, antigüedad de contrato (años completos), años activo (años con al menos un día en carretera), número medio de paradas por año activo, número medio de préstamos por año activo, y porcentaje de préstamos no devueltos (con respecto al total operados por ese empleado).

1.2.- Operatividad (2.5 puntos, 1+1+0.5 respectivamente)

Crear el *package foundicu* conteniendo los siguientes procedimientos (públicos):

- Procedimiento para insertar *Préstamo*: recibe una *signatura* (*signature*); verifica que el USER actual existe, y que hay una reserva en el ejemplar especificado (*signature*) para el usuario actual empezando con fecha de hoy; si esos requisitos se cumplen, convierte esa reserva en un préstamo. En caso contrario (sin reserva), verifica que la copia esté disponible durante dos semanas (desde la fecha actual), que el usuario actual no haya alcanzado el límite superior para préstamos (contando reservas), y que no esté sancionado; si el préstamo es posible, inserta una nueva fila de préstamo, y de lo contrario informa sobre el problema.
- Procedimiento para Insertar *Reserva*: recibe un ISBN y una fecha; verifica que el USER actual exista y que tenga cupo disponible para reservar (que no haya alcanzado el límite superior de préstamos y que no esté sancionado); verifica la disponibilidad de una copia de esa edición durante dos semanas (14 días) a partir de la fecha proporcionada y luego coloca (inserta) la reserva. De lo contrario, informa acerca del impedimento encontrado.
- Procedimiento para registrar *Devolución*: recibe una *signatura*; verifica que el USER actual tiene prestado ese libro y registra su devolución (de lo contrario, informa sobre el impedimento).

1.3.- Diseño Externo: perfil '*usuario de servicios*' (2.5 puntos, 0.5+1+1 resp.). Para esta sección, debe definirse el "usuario actual". Es elegible tomar el valor proporcionado por la función nularia USER (insertando ese "usuario" para realizar pruebas); o bien introducir en el paquete de la sección anterior (1.2) una

variable pública (por ejemplo, `current_user` CHAR(10), con un procedimiento para asignar y una función para devolver 'usuario actual').

- vista **my_data** (*read only*): contiene los datos personales del usuario actual (identificación, nombre completo, dirección, ...)
- vista **my_loans** (*operable*): muestra todos los préstamos (hasta la fecha) del usuario, junto con sus publicaciones si las hubiera (si no hay publicaciones sobre un préstamo, se devolverán valores nulos). Además de la consulta, la vista debe permitir actualizar el atributo 'post': al hacer esto, el atributo `post_date` se asigna automáticamente con la fecha y hora actuales; si no existía un post anterior (el valor anterior del post era nulo) los atributos 'likes' y 'dislikes' se inicializan a cero. No se puede actualizar (manualmente) ninguna otra columna. Tampoco se permiten borrados ni inserciones sobre esta vista.
- vista **my_reseervations** (*operable*): muestra todas las reservas de los usuarios actuales. Permite insertar filas, eliminar filas, y actualizar las fechas de una fila (siempre que el libro o cualquier otra copia del mismo ISBN esté disponible).

1.4.- Bases de Datos Activas / Disparadores (3 puntos, 1 pt c/u)

Diseñar, implementar y probar una solución para **tres** de los siguientes cuatro problemas con disparadores*:

- Evitar los "posts" de usuarios institucionales (bibliotecas municipales).
- Cuando el estado de una copia se establece como "deteriorado", la "fecha de baja" se establece automáticamente en "fecha y hora actuales".
- Crear "tablas de históricos" tanto para *usuarios* como para *préstamos* (no vistas, sino otras dos tablas idénticas). Cuando se elimina un usuario, crear un registro histórico de ese usuario y mover todos sus préstamos al histórico de préstamos.
- Añadir una columna "número de lecturas" (lecturas) a la tabla 'Libros'. Cuando se presta un libro, se actualizará el número de lecturas (+1).

(*) Nota: se deben elegir tres problemas para resolver (descartar uno); cada uno de estos problemas requerirá la creación de al menos un disparador (pero se pueden crear varios si se estima necesario).

2. – PUNTO DE PARTIDA

Para el desarrollo de esta práctica, se dispondrá de una base de datos completamente nueva, ya poblada con los datos provenientes de la base de datos original. Para replicar esta base de datos se proporcionarán scripts de creación y carga, junto con una breve documentación sobre este diseño (grafo relacional, y un subconjunto de los comentarios semánticos más relevantes).

El primer paso para llevar a cabo este trabajo es por tanto ejecutar estos scripts para replicar el entorno de desarrollo (recreando esta base de datos en su cuenta Oracle DB).

3. – MATERIAL DE APOYO

Además de clases y tutorías, los alumnos cuentan con los siguientes recursos:

- Recursos documentales:
 - Enunciado de la práctica (este documento).
 - Diapositivas de clase.
 - Solución de la primera práctica (grafo relacional y comentarios).
 - Plantilla para la redacción de la memoria (en formato docx).
- Recursos audiovisuales: clases en vídeo para adquirir los conocimientos específicos acerca del uso de las herramientas que se usarán en los laboratorios (sintaxis pl/sql) en forma de clase invertida.
- Recursos Sw:
 - Cuenta de usuario sobre el SGBDR Oracle (accesible desde las aulas y desde [aula virtual](#)), con privilegios para todas las operaciones necesarias y con acceso a las tablas de la BD obsoleta.
 - Script de creación que implementa la nueva estructura (tablas de la BD).
 - Script de migración de datos desde la BD obsoleta a la nueva BD.

4. APARTADOS A REALIZAR

Todos los resultados de esta práctica (diseños, código, pruebas) se recogerán en un solo documento (**memoria de prácticas**, en formato PDF). Para cada sección del enunciado establezca un capítulo en la memoria. Dentro de cada capítulo, incluya un apartado independiente por cada elemento requerido en el enunciado. Estos apartados deberán seguir el mismo orden que ha sido establecido en el enunciado.

Cada uno de estos apartados se presentará con la descripción del elemento a desarrollar (enunciado), y su resolución con tres subapartados, como sigue (puntuación entre paréntesis): **diseño** (30%), **implementación** (40%), y **pruebas** (30%). Obsérvese que descuidar cualquiera de estas partes puede implicar una severa pérdida de puntuación.

- a) **Diseño**: las consultas deberán describirse en álgebra relacional; hay que describir la lógica de los bloques de código pl/sql; y debe justificarse la parametrización de las reglas ECA (disparadores). Acompañarán al diseño los pertinentes comentarios acerca de la semántica implícita incorporada o la explícita no reflejada (allá donde fuera necesario). En su caso, deben documentarse cualquier modificación/creación de tablas, o la observación de nuevos supuestos semánticos implícitos.

- b) **Implementación:** código PL/SQL que implementa ese elemento. El código deberá recogerse en formato texto, de modo que sea posible trasladarlo a la consola sql*plus con operaciones de copia-pegar (copy-paste) y debidamente sangrado ('indentado' con espacios en blanco) para mejorar su legibilidad. También se comentará cualquier aspecto no incluido en el apartado anterior (diseño).
- c) **Pruebas:** descripción de las acciones a realizar para comprobar el correcto funcionamiento del elemento implementado, descripción del resultado esperado, y descripción del resultado obtenido (se puede acompañar de capturas de pantalla para ilustrar ese resultado). En caso de que el resultado obtenido difiera del esperado, deberá explicarse esa desviación, el problema detectado (en su caso) y las acciones necesarias para corregirlo.

Ejemplos:

- Una consulta puede comprobarse planificando su ejecución sobre distintos estados de la BD, estableciendo anticipadamente los resultados esperados y comparándolos con los resultados obtenidos. Las coincidencias se documentarán como pruebas fallidas (no detecta errores); y las divergencias como pruebas fructíferas, señalando las posibles causas y soluciones. En esta práctica no se solicitan pruebas exhaustivas, solo es necesario incluir algunos ejemplos de pruebas. Pero es importante observar que la simple ejecución de la consulta no es una prueba en sí misma.
- Una vista se prueba como una consulta.
- Un bloque puede comprobarse planificando ejecuciones con distintos parámetros, anticipando resultados esperados y comprobando su funcionamiento y efectos.
- Un disparador puede comprobarse provocando su activación en distintos casos, anticipando resultados esperados, y comprobando su funcionamiento y efectos. Debe comprobarse si produce tabla mutante (operando varias filas).

Documentar todo el trabajo realizado mediante la correspondiente **Memoria de Prácticas**, para lo cual se proporciona una plantilla para su redacción. Además de incluir los elementos solicitados con sus etapas de desarrollo (diseño, implementación, prueba), hay que asegurarse de que todas las decisiones de diseño estén convenientemente justificadas y así reflejadas en la memoria. Se guardará el documento como archivo **.pdf**, y se etiquetará **nia1_nia2_nia3_LW2.pdf** antes de enviarlo a través de Aula Global.

Sólo debe entregar un miembro de cada grupo de prácticas

Fecha límite: 4 de abril de 2025