PROYECTO FINAL – BASE DE DATOS

Con el siguiente esquema de tablas:

Carreras = (**id_carr**, nom_carr, reg_calif)
Estudiantes = (**cod_e**, nom_e, dir_e, tel_e, fech_nac, *id_carr*)
Asignaturas = (**cod_a**, nom_a, int_h, creditos)
Profesores = (**id_p**, nom_p, dir_p, tel_p, profesion)
Libros = (**isbn**, titulo, edicion, editorial)
Autores = (**id_a**, nom_autor, nacionalidad)
Escribe = (***isbn***, ***id_a***)
Ejemplares = (***isbn***, **num_ej**)
Imparte = (***id_p***, ***cod_a***, ***grupo***, horario)
Inscribe = (***cod_e***, ***cod_a***, ***id_p***, ***grupo***, n1, n2,
n3) Presta = (***cod_e***, ***isbn***, ***num_ej***, **fech_p**,
fech_d) Referencia = (***cod_a***, ***isbn***)

Es necesario construir una base de datos en la universidad para la **facultad de ingeniería**, que cumpla con los siguientes requerimientos:

La facultad de ingeniería de la universidad será la responsable de mantener y administrar la información de su competencia, esto es, sus estudiantes y docentes. Se buscará poder administrar de la forma más eficiente posible la información, buscando que cada **proyecto curricular**, también tenga su independencia de la mejor forma posible. Se tendrá un esquema para ingeniería así como los esquemas para cada programa de ingeniería (los estudiantes son gestionados por cada programa curricular).

Nota aclarativa: los profesores tienen un contrato firmado con la facultad, y como tal pueden impartir asignaturas en cualquier programa ofrecido por la facultad a la que pertenece. No manejaremos los casos en los que profesores imparten en otras facultades (de hecho, en este caso, no existen más facultades).

La biblioteca se asume como única en la facultad, independientemente de las sedes o edificaciones donde esta funciona.

La facultad cuenta con un servidor, y la base de datos que usa está en el DBMS PostgreSQL.

Se deberán definir los siguientes **grupos de usuarios**, y los usuarios creados se asignarán a cada grupo apropiadamente para que, junto con el esquema descrito, se configuren los siguientes niveles de acceso y funciones así:

Decano (vamos a usar el usuario admin "postgres" y asumir que es el decano/coordinador)

- Consultar el listado de los estudiantes de la facultad (nombre y código) y las asignaturas (nombre, código y grupo) que cursan (listado_facultad_asig).
- Consultar el listado de los estudiantes de la facultad (nombre y código) y las asignaturas (nombre, código y grupo) con sus respectivas notas (listado_facultad_notas).
- Ver la información de los libros prestados por la biblioteca (prestamos_universidad).

Profesor

- Consulta la lista de **sus** estudiantes (lista_estudiantes) con información de las asignaturas, del estudiante y las notas.
- Actualiza notas de la lista de **sus** estudiantes.

- Consulta libros, autores y quien los escribe (consulta_escribe).
- Puede consultar su información como profesor y actualizarla (info_profesores).

Estudiante

- Puede consultar sus notas (notasEstud).
- Consulta libros, autores y quien los escribe (consulta_escribe).
- Puede consultar sus préstamos, incluyendo fecha de préstamo y devolución (consultar_prest_est).

Bibliotecario

- Administra la información de los ejemplares (insert, update, delete, select).
- Puede ingresar libros y sus autores (insert, update, delete, select).
- **PUNTO DE BONO (No es obligatorio):** Administra la información de los préstamos de libros (insert, update, select).

Se trabajará la implementación del esquema de tablas (DDL), la definición de las vistas que considere indispensables para lograr la funcionalidad requerida por parte de los diferentes usuarios (DML), la creación de los roles y usuarios necesarios y los permisos o accesos de ellos a lo necesario para sus consultas y operaciones sobre la base de datos (DCL).

Adjunto encuentra el archivo **facultadIngenieria-guia.sql**. En ese script está la serie de pasos en el orden debido que el equipo debe desarrollar. Note que en el fichero hay indicaciones y SQL parcial que, según lo solicitado, deberá ser complementado para satisfacer lo necesitado.

Deben crearse al menos 4 programas curriculares (carreras). Los estudiantes se ingresan por carrera, es decir, cada esquema de programa tiene su tabla “estudiantes”. La biblioteca es otro esquema con sus tablas. Las vistas deben responder a la necesidad de las consultas; en el fichero encuentra las indicaciones y sintaxis para la creación de vistas. Por facilidad se sugiere que las vistas lleven los nombres que se han ubicado entre paréntesis en lo ítems anteriores. Las vistas esperadas con los nombres solicitados son:

- info_profesores.
- estudiantes_fac.
- lista_estudiantes.
- notasEstud.
- consultas_escribe.
- consultar_prest_est.
- listado_facultad_asig.
- listado_facultad_notas.
- prestamos_universidad.

La carga de datos debe ser masiva. Use sentencias para cargar desde archivos CSV. Use los datos que fueron remitidos en la primera parte del proyecto (Datos Academia). Deben existir al menos 4 estudiantes por carrera, es decir que si necesita crear más, asignar asignaturas, prestar libros, etc., según considere, deberá indicarlo al momento de sustentarlo.

La creación de los usuarios, y sus respectivos permisos y accesos se deberán realizar a partir de un script que deben presentar, dónde cada uno de los usuarios registrados (estudiantes, bibliotecario y profesores) tendrán su usuario y contraseña, así como el

respectivo rol. La contraseña de cada usuario será creada igual que el nombre del usuario en la base de datos. En el script de guía **ya se le está apoyando con la creación de roles** para estudiantes y profesores, parcialmente.

En el archivo guía encontrará la sección "triggers y procedimientos"; esta debe ejecutarse tal cual está sin ninguna modificación; es una **ayuda** que le brinda el profesor para crear los usuarios automáticamente a través de funciones y acciones adicionales para soportar algunas consultas. Si desea crear los usuarios manualmente, omita esta parte.

Adicionalmente, encuentra el adjunto **operaciones-auxiliares.sql**. Allí encuentra sentencias SQL de ayuda para verificar roles, eliminar y revocar privilegios en caso de requerirlo y un método para eliminar usuarios por si se ha equivocado al hacer pruebas o demás.

Los archivos para remitir son:

- SQL completo: **proyecto-ing-grupo-00.sql**. Este fichero debe tener el DDL con el que creó los esquemas, las tablas con restricciones, las vistas, la carga de datos, los roles, usuarios y configuración de permisos. **El fichero se ejecuta el día de la sustentación y debe pasar sin problemas; en caso de fallo la calificación es cero (0.0). Revise antes de la sustentación que le funciona. Para esto el profesor le está dando un paso a paso, comentarios e incluso SQL del proyecto.**
- SQL de pruebas: Debe crear un fichero **pruebas-ing-grupo-00.sql** con los ejemplos para probar en la sustentación para cada uno de los roles. Básese en el adjunto **pruebas-guia.sql** para construir el suyo.
- Si existen modificaciones al esquema de la base de datos, también se deben incluir.

La sustentación equivale al 70% del corte final. El 40% es parte grupal del proyecto y el 30% es parte individual. La matriz de calificación (rúbrica) se encuentra adjunta a la tarea de entrega de ficheros para dejar evidencia. Solo se pueden demorar a lo sumo 20 minutos en la sustentación. El fichero solicitado se ejecuta y debe pasar sin problemas, de no ser así tendrá calificación **cero (0.0)**; con el fichero de pruebas se solicitará al equipo que muestre la funcionalidad adecuada de acuerdo con lo solicitado. Para la parte individual a cada miembro se le hará una pregunta y/o solicitud de consulta y su realización será evaluada con cumple (máximo de puntaje 5 o 50 puntos) o no cumple (mínimo del puntaje 0 puntos).