

Carrera: INGENIERIA EN INFORMATICA

Asignatura: 3641 – Bases de Datos Aplicada.

Tema: Repaso de SQL

Unidad: 1

Objetivo: Repasar los conceptos básicos de SQL. Profundizar su comprensión y sumar nuevas herramientas. Que el alumno sea capaz de resolver problemas complejos pudiendo optar por distintos métodos según su criterio.

Competencia/s a desarrollar:

- Especificación, proyecto y desarrollo de sistemas de información.
- Especificación, proyecto y desarrollo de sistemas de comunicación de datos.
- Especificación, proyecto y desarrollo de software.
- Proyecto y dirección en lo referido a seguridad informática.
- Establecimiento de métricas y normas de calidad de software.
- Procedimientos y certificaciones del funcionamiento, condición de uso o estado de sistemas de información, sistemas de comunicación de datos, software, seguridad informática y calidad de software.
- Dirección y control de la implementación, operación y mantenimiento de sistemas de información, sistemas de comunicación de datos, software, seguridad informática y calidad de software.
- Identificación, formulación y resolución de problemas de ingeniería en sistemas de información/informática.
- Concepción, diseño y desarrollo de proyectos de ingeniería en sistemas de información / informática.
- Gestión, planificación, ejecución y control de proyectos de ingeniería en sistemas de información / informática.
- Utilización de técnicas y herramientas de aplicación en la ingeniería en sistemas de información / informática.
- Generación de desarrollos tecnológicos y/o innovaciones tecnológicas.
- Desempeño en equipos de trabajo.
- Comunicación efectiva.

Descripción de la actividad:

1. Tiempo estimado de resolución: 1 semana
2. Metodología: En computadora.
3. Forma de entrega: No obligatoria
4. Metodología de corrección y feedback al alumno: Presencial y por Miel.

Pauta general: Todos los ejercicios deben resolverse con DML/DDL. Aunque puede usar la GUI en caso de no recordar la sintaxis, recomendamos enfáticamente practicar y aprender la sintaxis de SQL.

Los ejercicios se pensaron para ser resueltos en Microsoft SQL Server. Puede usar la versión Developer o Express. Recomendamos emplear versión 2019 o 2022. Sin embargo, si lo desea puede resolverlos en PostgreSQL o MySQL.

1. Cree una base de datos con los valores por defecto de crecimiento, ubicación de archivos, etc. Llame a la misma con su nombre o el de su grupo (si opta por resolver el TP en grupo).
2. Cree un esquema denominado “ddbba” (por bases de datos aplicada). Todos los objetos que cree a partir de aquí deberán pertenecer a este esquema o a otro según corresponda. No debe usar el esquema default (dbo).
3. Cree una tabla llamada “registro” que tenga los siguientes campos: id entero autoincremental como primary key; fecha y hora con valor default del momento de inserción, texto con un tamaño máximo de 50 caracteres, modulo con un tamaño máximo de 10 caracteres. Esta tabla la empleará como bitácora (log) de las operaciones de los puntos siguientes.
4. Cree un SP llamado “insertarLog” que reciba dos parámetros: modulo y texto. Si el modulo llega en blanco debe utilizar el texto “N/A”. Este SP insertará registros en la tabla de bitácora. A partir del siguiente punto, cada inserción, borrado, modificación, debe crear un registro en la mencionada tabla. En ningún caso debe realizar INSERTs por fuera del SP insertarLog.
5. Modele la relación entre persona, curso, materia. Una materia tiene varios cursos. Un curso tiene varias personas. Una persona puede ser alumno o docente en cada materia (no ambos al mismo tiempo), pero puede ser docente en una y alumno en otra. Genere tablas para cada uno. Asegúrese de aplicar los siguientes conceptos:
 - a. Claves primarias en cada tabla
 - b. Claves foráneas (restricciones) para vincular cada tabla a las demás.
 - c. Las personas pueden opcionalmente tener un vehículo. Incluya la patente como campo de carga opcional con las restricciones de verificación correspondientes.
 - d. Los cursos (comisiones) tienen un número de comisión de cuatro dígitos.
 - e. Las personas tienen un DNI y un número de teléfono, también una localidad de residencia y una fecha de nacimiento. Nombres y apellido se almacenan por separado.
 - f. Las materias deben tener un campo ID autoincremental como PK. Ese campo lo debe completar el DBMS en cada inserción.

- g. Los aspectos de diseño que no estén detallados se dejan a su criterio.
Documente las decisiones en la forma de comentarios en los scripts.
6. Compruebe que las restricciones creadas funcionen correctamente generando juegos de prueba que no se admitan. Documente con un comentario el error de validación en cada caso. Asegúrese de probar con valores no admitidos siquiera una vez cada restricción.
 7. Cree un stored procedure para generar registros aleatorios en la tabla de alumnos. Para ello genere una tabla de nombres que tenga valores de nombres y apellidos que podrá combinar de forma aleatoria. Al generarse cada registro de alumno tome al azar dos valores de nombre y uno de apellido. El resto de los valores (localidad, fecha de nacimiento, DNI, etc.) génere los en forma aleatoria también. El SP debe admitir un parámetro para indicar la cantidad de registros a generar.
 8. Utilizando el SP creado en el punto anterior, genere 1000 registros de alumnos.
 9. Elimine los registros duplicados utilizando common table expressions.
 10. Cree otro SP para generar registros aleatorios de comisiones por materia. Cada materia debe tener entre 1 y 5 comisiones, entre los distintos turnos.
 11. Genere un SP para crear inscripciones a materias asignando alumnos a comisiones.
 12. Cree una vista empleando la opción "SCHEMABINDING" para visualizar las comisiones (nro de comisión, código de materia, nombre de materia, apellido y nombre de los alumnos). El apellido y nombre debe mostrarse con el formato "Apellido, Nombres" (observe la coma intermedia).
 - a. Verifique qué ocurre si intenta modificar el tamaño de uno de los campos de texto de la tabla de alumnos.
 - b. Verifique qué ocurre si intenta agregar un campo a la tabla de alumno.
 - c. Verifique qué ocurre si intenta agregar un campo que admita nulos a la tabla de alumno. ¿Hay diferencia entre agregarlo si la tabla está vacía o tiene registros?
 - d. ¿Puede usar SCHEMABINDING con una vista del tipo "Select * From.."?
 13. Agregue a la tabla de comisión soporte para día y turno de cursada. (Modifique la tabla). Los números de comisión son únicos para cada cuatrimestre.
 14. Complete los datos de día y curso con valores aleatorios.
 15. Genere una función validaCursada que devuelva la cantidad de materias superpuestas a las que está inscripto un alumno, recibiendo el DNI por parámetro.
 16. Cree una vista que utilice la función del punto anterior y muestre los alumnos con superposición de inscripciones.
 17. Cree un SP que elimine las inscripciones superpuestas o duplicadas.

18. Cree una vista que presente una vista PIVOT de cantidad de inscripciones para las materias por cada turno. Utilice su criterio para presentarlo de la manera que considere más clara.
19. Utilizando Window Functions cree una vista que muestre los alumnos inscritos a una materia y en una columna también muestre la cantidad total de materias a las que se ha inscrito ese alumno (en un mismo cuatrimestre).
20. Utilizando Window Functions presente el 5% más joven y el 5% menos joven del alumnado.

Guarde periódicamente el script. Al finalizar genere un backup de la base de datos.