

CASO DE PRÁCTICA - Gestión de proyectos

La empresa **TechSolutions** necesita implementar un sistema informático para organizar los proyectos de desarrollo que gestiona. El objetivo es tener un control ordenado de los proyectos en curso, las tareas involucradas y la asignación de cada empleado dentro de los equipos de trabajo.

Cada proyecto tiene un nombre, una fecha de inicio y opcionalmente una fecha de fin. Un proyecto puede incluir muchas tareas.

Los empleados se registran con nombre y correo electrónico (el correo es único). Un empleado puede participar en varios proyectos, y en cada proyecto asume un único rol específico (por ejemplo: Director de Proyecto, Analista Funcional, Desarrollador, Tester, Diseñador). El sistema debe garantizar que en un proyecto un empleado sólo pueda aparecer una vez, con un rol definido. Cada asignación de un empleado dentro de un proyecto incluye la fecha de inicio de la participación y, en caso de corresponder, la fecha de fin.

A su vez, cada proyecto se descompone en tareas. Una tarea tiene una descripción, fechas de inicio y fin, y un estado (ejemplo: Pendiente, En progreso, Finalizada). Cada tarea se identifica dentro de su proyecto mediante un número de orden.

Actividad

Diseñar un modelo de datos correctamente, aplicando los principios de normalización, consistencia y restricciones, e implementar el diseño físico en SQL para SQL Server siguiendo buenas prácticas de nomenclatura.

Metodología del examen

1. Entrega de un script SQL de la implementación del modelo normalizado. Tiempo: 40 minutos
2. Cuestionario en Moodle. El cuestionario se habilita luego de la entrega del script sql. Tiempo: 20 minutos

Criterios de evaluación

1. **Diseño de la Base de Datos.** Interpretar el caso de estudio y reconocer entidades fuertes, débiles y asociativas. Identificar relaciones 1:N y M:N.
2. **Modelo relacional y normalización.** Derivar correctamente tablas a partir del diseño conceptual. Aplicar normalización hasta 3FN (o BCNF si corresponde).
3. **Consistencia y no redundancia.** Evitar redundancias en la estructura. Proponer tablas de clasificación o tipificación para evitar duplicación de datos.
4. **Claves primarias y foráneas.** Definir PK y FK siguiendo las reglas del modelo relacional. Identificar correctamente las claves compuestas.
5. **Implementación física en SQL.** Traducir el diseño al modelo físico en SQL Server, usando convención *snake_case*, ajuste de tipos de datos y restricciones NULL y NOT NULL.
6. **Restricciones.** Incluir restricciones con CHECK, UNIQUE y DEFAULT

Consideraciones

- Usar nombres en *snake_case* (ejemplo: usuario_id, libro_autor).
- Declarar restricciones con control de nombres

