Trabajo Practico Modelo Relacional

Alumnas: Julia Bonetto, Celina Maldonado, Priscila Restivo
Curso: 6to Informatica
Profesor: Marcelo Acevedo
Materia: Base de Datos

Funcionalidades: alojamiento de repositorios, seguimiento de versiones, gestión de problemas (issues)

Actividades

● Armen grupos de no más de 4 integrantes.

● Elijan una aplicación o página web, como por ejemplo:Mercado Libre,Spotify u otra que conozcas.

● Seleccionen entre dos y cuatro funcionalidades principales que tenga esa app. Ejemplos: “Publicar un producto” y “Comprar un producto”,“Crear una playlist” y “Seguir un usuario”

● Analicen qué entidades están involucradas en esas funcionalidades. Preguntas de guía: ¿Quiénes son los usuarios? ¿Qué objetos se gestionan? (productos, canciones, pedidos, videos…) ¿Qué relaciones hay entre ellos?

● Realicen un modelo Entidad-Relación (E-R)

● Pasen ese modelo E-R al modelo relacional.

Grafico en Draw.io:
https://app.diagrams.net/#G1tlz-TwwPL_UUF3ymlKzLl2EZB4rjm67z#%7B%22pageId%22%3A%22mqEPi1Oz0cv9EyF-Dl6t%22%7D

Alojamiento de Repositorios

Esta funcionalidad permite a los desarrolladores almacenar y gestionar sus proyectos de código de fuente de manera centralizada. Son lugares en la nube en donde se guardan los archivos y las distintas versiones de este. Te permite trabajar de manera colectiva o individual.

Entidades Modelo Relacional:

Usuario
Atributos:
ID usuario
Nombre
Email
Rol(desarrollador, colaborador).

Repositorio
ID repositorio
Nombre
Visibilidad(pública o priv)
PK url
FK ID_usuario

Pull request
ID pull request
Título
Estado (abierto, cerrado o fusionado)
Fecha de creación
FK ID usuario
FK ID repositorio

Colaborador
FK ID usuario
FK ID repositorio
Rol(escritor, lector o editor)

Organización
ID organización
PK url
Nombre

Relaciones:
Usuario-Repositorio 1:N: Un usuario puede tener muchos repositorios.
Repositorio-Problema 1:N: Un repositorio puede tener muchos problemas.
Usuario-PullRequest 1:N: Un usuario puede crear muchos pull requests.
Organización-Repositorio 1:N: Una organización puede tener muchos repositorios.
Usuario-Organización N:M: Un usuario puede ser miembro de varias organizaciones y una organización puede tener varios usuarios.
Repositorio-Colaborador N:M: Un repositorio puede tener varios colaboradores y un usuario puede colaborar en varios repositorios.

Entidades Modelo Entidad Relación:

Usuario
Atributos:
Rol
Nombre
Email

Pull request
Atributos:
Estado
Título
Fecha de creación

Repositorios
Atributos:
Modo(individual u organización)
Nombre
Visibilidad
Url

Relaciones:
Usuario-Repositorio N:M: Un usuario puede tener muchos repositorios, y un repositorio puede tener muchos usuarios.
Usuario-Pull Request 1:N: Un usuario puede crear muchos pull request.
Problema-Repositorios 1:N: Puede haber muchos problemas dentro de un repositorio.

Seguimiento de Versiones

Función: Permite registrar y gestionar los cambios realizados en el código fuente de un proyecto a lo largo del tiempo. Cada modificación se guarda con un identificador único, asociado a un autor, fecha y mensaje descriptivo.

Entidades Modelo Relacional

Commit
Atributos:
Mensaje
Fecha
FK ID usuario
FK ID repositorio

problema-commit (Tabla intermedia para relacionar muchos commits con muchos issues)

Atributos:

FK ID problema
FK ID commit

Entidades Modelo Entidad Relación

Commit
Atributos:
Mensaje
Fecha

-Relaciones:
Repositorio → Commit: Un repositorio puede tener muchos commits.
Usuario → Commit: Un usuario puede realizar muchos commits.
Commit ↔ Issue: Un commit puede resolver muchos issues y un issue puede estar relacionado con muchos commits (relación muchos a muchos, representada por la tabla intermedia problema-commit).

Gestión de problemas (issues)

Su función es realizar un seguimiento de tareas e hitos, así como para vincular incidencias con solicitudes de incorporación de cambios y confirmaciones. Facilita la comunicación de errores o problemas que se encuentren en el código, permitiendo a los desarrolladores abordar y solucionar esos problemas.

Entidades Modelo Relacional:

Usuario
Atributos:
PK-ID usuario
Nombre
Email
Rol(desarrollador, colaborador).

Problema
Atributos:
PK-ID problema
Título
Descripción
Estado (abierto, cerrado),
Prioridad (alta, media, baja)
Fecha creación
FK-ID usuario
FK-ID repositorio

Comentario
Atributos:
PK-ID comentario
Comentario
Fecha
FK-ID usuario
FK-ID problema

Etiqueta
Atributos:
PK-ID etiqueta
Nombre etiqueta
FK-ID comentario

Comentar
Atributos:
FK-ID usuario
FK-ID comentario

Relación:
Usuario-Problemas 1:N: Un usuario puede crear muchos problemas
Problema-Comentario 1:N: Un problema puede tener múltiples comentarios
Problema-Etiqueta 1:N: Un problema puede incluir muchos etiquetas
Usuario-Comentario N:M: Los usuarios pueden comentar muchos problema (tabla intermedia “comentar”)

Entidades Modelo Entidad Relación:

Usuario
Atributos:
Nombre
Email
Rol(desarrollador, colaborador).

Problema
Atributos:
Título
Descripción
Estado (abierto, cerrado),
Prioridad (alta, media, baja)
Fecha creación

Comentario
Atributos:
Comentario
Fecha

Etiqueta
Nombre etiqueta

Relación:
Usuario-Problemas 1:N: Un usuario puede crear muchos problemas
Usuario-Comentario N:M: Un usuario puede hacer muchos comentarios
Problema-Etiqueta 1:N: Un problema puede tener muchas etiquetas
Comentario-Etiqueta 1N:N Un comentario tiene muchas etiquetas