

Questão 1: O que são sistemas de controles de versões? Por que são importantes?

São sistemas que possibilitam guardar snapshots de seu projeto ao longo do tempo. E são importantes pois abrem a possibilidade de "voltar no tempo" no projeto e visitar e até implementar uma versão anterior caso seja necessário .

Questão 2: Qual a diferença entre o Git e o GitHub? Como eles estão relacionados? É possível usar um sem o outro?

Não, o Git é um sistema LOCAL de controle de versões. E a diferença básica entre eles é como os desenvolvedores comunicam as alterações feitas entre o time.

Questão 3: O Git é um sistema distribuído de controle de versões. O que significa isso?

Git é o sistema de controle de versões, já o GitHub é uma plataforma Web que incorpora o Git e facilita trabalhos colaborativos. O GitHub incorpora o Git adicionando novas features como pull, fork e clone para facilitar o trabalho conjunto. É possível usar o Git sem o GitHub mas não o contrário.

Questão 4: Existem pelo menos dois erros no projeto do Elmasri envolvendo as tabelas funcionário e departamento. Identifique quais são esses erros, explique o que está errado e conserte os erros se for necessário.

O relacionamento entre as tabelas é de 1:N. No entanto, um único departamento não pode ter mais de um funcionário como seu gerente. Logo, o relacionamento correto seria de 1:1, em que um gerente trabalharia em um único departamento e o departamento teria apenas um funcionário responsável.

Na tabela funcionário, o campo "cpf_gerente" deve permitir valores nulos, visto que nem todos os funcionários, como o diretor geral por exemplo, possuem supervisores.

Questão 5: Alguma tabela do projeto representa um relacionamento do tipo N:N? Se sim, identifique a tabela e explique porque é um relacionamento N:N; Se não, explique porque não há relacionamentos N:N neste projeto.

Sim existe, é o relacionamento entre "funcionário" e "projeto". E é um relacionamento N:N pois "N" funcionários podem trabalhar em "N" projetos, e esse relacionamento gera a tabela "trabalha em" contendo as chaves primarias das duas tabelas "pai".

Questão 6: Pelo projeto apresentado é possível que um determinado funcionário seja o gerente de um ou mais departamentos. Se eu quisesse impor uma regra que diz que um funcionário só pode ser gerente de, no máximo, um departamento, o que eu teria que fazer no projeto? O que eu teria que criar no banco de dados para impor essa restrição? Por quê?

No projeto, basta mudar o relacionamento entre “funcionário” e “departamento” de 1:N para 1:1. No SQL, adicionamos na criação da tabela “UNIQUE (cpf_gerente)”. Desta forma, todo gerente só conseguirá ser cadastrado uma única vez em um departamento.

Questão 7: Por que o relacionamento entre as tabelas departamento e projeto está representado como uma linha pontilhada? O que isso representa? Por que foi representado assim?

Trata-se de um relacionamento não identificado, em que a chave primária da tabela “departamento” é uma chave estrangeira que não compõe a chave primária da tabela projeto. É representada dessa maneira para diferenciar dos relacionamentos identificados.

Questão 8: Qual é o único tipo de relacionamento que pode guardar dados? Por quê? Existe algum relacionamento assim no projeto do Elmasri?

O relacionamento N:N pois ele gera uma tabela para armazenar os dados das duas tabelas pai. Esse tipo de relacionamento é encontrado entre “funcionário” e “projeto”.

Questão 9: Como explicar o relacionamento da tabela funcionário com ela mesma? É um erro? É correto? Por quê?

É correto, pois é um relacionamento em que o funcionário torna sua chave primária em uma chave estrangeira, para ser o supervisor de um funcionário comum, pois um supervisor ainda é um funcionário, então ele ainda se enquadra na mesma tabela.

Questão 10: Qual a diferença entre banco de dados, usuário e esquema no PostgreSQL?

Um usuário do SGBD com permissões administrativas pode criar um banco de dados. Os esquemas guardam os metadados e o catálogo de dados das tabelas que formam os bancos de dados. Logo, um usuário possui poder para criar um banco de dados e criar esquemas dentro do banco de dados.

Questão 11: Por que um esquema é importante?

O esquema criado é importante para separar os metadados e objetos que utilizamos no projeto dos outros utilizados pelo esquema público.

Questão 12: Se você não definir um esquema específico, onde os objetos do banco de dados (tabelas, relacionamentos, dados, etc.) serão gravados? Isso é bom ou ruim? Por quê?

Eles iriam para o esquema ‘public’. Isso é ruim, pois os objetos iriam ser salvos junto com os objetos já existentes no esquema público, dificultando a organização e as operações dentro do esquema.

Questão 13: Agora que você já implementou o projeto no PostgreSQL, tem alguma sugestão de melhoria a fazer para o projeto? Como ele poderia ser melhorado?

Aumentar a quantidade máxima de caracteres nos campos de texto, visto que os endereços e os locais podem possuir muitos caracteres.