

Lógica

1. Escreva uma função que recebe como entrada três números inteiros maiores que zero que representam os lados de um triângulo e que deve retornar como resposta que tipo de triângulo é aquele. Caso os valores informados não sejam de um triângulo válido, por exemplo quando a soma de dois lados é menor que o terceiro, a função deve informar isso para o usuário.

Nome de Triângulo	Descrição
Escaleno	Possui todos os três lados diferentes
Isósceles	Possui dois lados iguais
Equilátero	Possui três lados iguais

2. Escreva uma rotina para exibir na tela os números de 1 a N. Para os múltiplos de 3, escreva FIZZ; para os múltiplos de 5, escreva BUZZ; e para os múltiplos de 3 e 5, escreva FIZZBUZZ.

3. Dado um vetor de 100 números inteiros, faça uma rotina que informa se há ou não números repetidos.

SQL

Considere as tabelas abaixo.

Professor (Codigo, Nome)
Aluno (Codigo, Nome)
Disciplina (Codigo, Nome)
Turma (Codigo, Professor, Disciplina, DataInicio, DataTermino)
Matriculado (Turma, Aluno, Nota1, Nota2, Nota3)

Obs.: campos em negrito são chaves primárias e campos sublinhados são chaves estrangeiras para as tabelas de mesmo nome.

1. Escreva uma consulta SQL que retorne código de todas as turmas cursadas pelo aluno de código 4 que tiveram seu início nos anos de 2009 e 2010. Informe também o nome do professor e o nome da disciplina de cada uma dessas turmas.
2. Escreva uma consulta SQL que retorne o nome e a nota final dos alunos que foram reprovados em alguma turma ministrada pelo professor 'Francisco Ferreira Chagas'. Veja as definições de 'nota final' e 'aprovação/reprovação' na questão 4.
3. Escreva uma consulta SQL que retorne o nome de todos os professores que nunca ministraram nenhuma disciplina.
4. Escreva uma consulta SQL que retorne nome do professor, nome da disciplina, data de início e data de término de todas as turmas que não tiveram nenhum aluno matriculado.
5. Escreva uma consulta SQL que retorne o código e nome de todos os alunos que já cursaram uma mesma disciplina pelo menos duas vezes. Informe também o nome da disciplina.

Aplicação: Controle de Abastecimentos

Requisitos funcionais:

Você foi contratado para analisar e desenvolver um software para gerenciar os abastecimentos e tanques de combustíveis do posto ABC. Sabendo disso, leia com atenção as informações que o gerente do posto descreveu:

1. Preciso controlar os abastecimentos feitos durante cada dia, identificando a bomba utilizada, a quantidade de litros e o valor abastecido;
2. Em cada abastecimento incide um imposto de 13% do valor abastecido, e essa informação deve ser registrada;
3. Cada bomba está ligada a um tanque. No posto ABC existem dois tanques, um de gasolina e um de óleo diesel. Para cada tanque duas bombas de combustível;
4. Também gostaria de um relatório em que os abastecimentos fossem agrupados, exibindo o dia, o tanque, a bomba e o valor. E ao final do relatório a soma total do período.

Requisitos não funcionais:

1. A aplicação deve ser simples (Minimum Viable Product);
2. Utilize boas práticas e conhecimentos que você julga interessantes;
3. Caso sinta necessidade crie testes unitários/funcionais;
4. A versão do Delphi fica a seu critério, mas utilize apenas componentes/recursos nativos;
5. Para construção do relatório utilize o [Fortes Report](#);

6. Utilizar banco de dados firebird ou sqlite.

Requisitos não funcionais que vão ser bem pontuados:

- 1.Tempo de Entrega.
- 2.Usar github para publicação do projeto
- 3.Usar Delphi XE6
- 4.Orientação a Objetos
- 5.Rtti, Helpers
- 6.Firedac