

Projeto:

Sistema de cadastro com busca de informações gerais.

Entregavel:

Front-End com formulário para input e controle dos dados.

Todos os procedimentos e instruções do SQL Server, devem ser feitas em PL-SQL, não podendo serem inseridas informações manualmente.

A conclusão projeto é a entrega de duas etapas, formulário de cadastro e visualização das informações do banco de dados.

A entrega deve ser feita em seu repositório GITHUB

Frameworks e Bibliotecas:

Você só poderá usar as bibliotecas e frameworks listados a seguir

- Node.js v11.0.0 (sevidor)
- Express (biblioteca para requisições)
- Body-Parser (receber informações de requisições)
- EJS (view render)
- JQuery
- Semantic UI (framework para front-end / CSS)
- Tedious (conexão com o banco de dados)
- Tedious-Promises (conexão com o banco de dados, porém com Promises)
- Axios ou Request (Requests de API)

Acessos:

Link API: <http://138.68.29.250:8082/>

Acesso ao banco de dados Microsoft SQL Server:

host: virtual2.febracorp.org.br:1433,
user: user_trial,
pass: 7412LIVE!@#\$%`&*()
Database: CONTOSO

1. Crie uma tela com um formulário com os seguintes campos:

- Nome
- Sobrenome
- E-mail

2. Envie os valores para o link da api colocando no campo data o JSON abaixo, com o método POST e no header o 'Content-Type' como 'application/x-www-form-urlencoded'

Modelo:

```
{  
  nome: 'nome',  
  sobrenome: 'sobrenome',  
  email: 'email'  
}
```

A API irá retornar um código como esse:

Modelo: "N#001#S#002#E#003#"

N=nome, S=sobrenome, E=email, #=separador de código

3. Use o resultado da API no exercício anterior, quebrando a string para obter os códigos, e conecte no banco de dados via backend (Node), para fazer um INSERT no banco, nas seguintes tabelas: obs. O id das tabelas é criado automaticamente, portanto não precisa ser inserido nenhum valor nesse campo.

Neste momento é necessário criar um método onde o código espere que os dados sejam inseridos para ter certeza de que os dados foram inseridos para o próximo exercício.

tbs_nome		tbs_sobrenome		tbs_email	
id	int	id	int	id	int
nome	nvarchar(100)	sobrenome	nvarchar(100)	email	nvarchar(100)
cod	bigint	cod	bigint	cod	bigint

4. Com os dados inseridos nas tabelas, use o código de cada um dos campos para fazer um SELECT nas tabelas a seguir para conseguir o campo “soma” de cada um dos códigos.

Ex:

Código do nome: 001	Código do sobrenome: 002	Código do email: 003
Soma do nome: 11	Soma do sobrenome: 22	Soma do email: 33

Tabelas do banco de dados:

tbs_cod_nome		tbs_cod_sobrenome		tbs_cod_email	
id	int	id	int	id	int
cod	bigint	cod	bigint	cod	bigint
soma	int	soma	int	soma	int

5. Para cada campo (nome, sobrenome, email) com seu respectivo código e soma, faça uma conta matemática de soma da coluna cod com a coluna soma para cada um dos casos (nome, sobrenome e email). Em seguida faça a soma entre todos os números resultantes das somas.

Nome: cod + soma
Sobrenome: cod + soma
Email: cod + soma

+
Total

Ex:

Código do nome: 001	Código do sobrenome: 002	Código do email: 003
Soma do nome: 11	Soma do sobrenome: 22	Soma do email: 33

Nome = 001 + 11

sobrenome = 002 + 22

Email = 003 + 33

Total = nome + sobrenome + email

6. Faça um SELECT no banco de dados com tabelas referência abaixo usando o total do exercício anterior, criando uma query usando JOIN para obter um animal, uma cor, e um país. E com LEFT JOIN retirar as cores excluídas dessa mesma query.

Obs: Use o campo total como chave, não o id.

tbs_animais	tbs_cores	tbs_cores_excluidas	tbs_paises																								
<table><tr><td>id</td><td>int</td></tr><tr><td>animal</td><td>nvarchar(100)</td></tr><tr><td>total</td><td>bigint</td></tr></table>	id	int	animal	nvarchar(100)	total	bigint	<table><tr><td>id</td><td>int</td></tr><tr><td>cor</td><td>nvarchar(100)</td></tr><tr><td>total</td><td>bigint</td></tr></table>	id	int	cor	nvarchar(100)	total	bigint	<table><tr><td>id</td><td>int</td></tr><tr><td>cor</td><td>nvarchar(100)</td></tr><tr><td>total</td><td>bigint</td></tr></table>	id	int	cor	nvarchar(100)	total	bigint	<table><tr><td>id</td><td>int</td></tr><tr><td>pais</td><td>nvarchar(100)</td></tr><tr><td>total</td><td>bigint</td></tr></table>	id	int	pais	nvarchar(100)	total	bigint
id	int																										
animal	nvarchar(100)																										
total	bigint																										
id	int																										
cor	nvarchar(100)																										
total	bigint																										
id	int																										
cor	nvarchar(100)																										
total	bigint																										
id	int																										
pais	nvarchar(100)																										
total	bigint																										

8. Mostre na tela onde o usuário irá preencher o formulário, o resultado. Que deve ser 1 animal, 1 cor e 1 país.