Olá Johnatas, tudo tranquilo?

Você está recebendo um teste na etapa de **Avaliação Técnica** do nosso processo seletivo para **Desenvolvedor Python/Django**. É de extrema importância que ele seja realizado por você, e que seja feito com a calma e atenção que merece esse processo seletivo.
Sendo assim, você deverá respondê-lo e nos retornar com o material salvo em PDF em até **2 (dois) dias** a contar da data de envio.

Não esqueça de colocar seu nome e a data que está enviando.

Boa Sorte!

Nome: Johnatas Rabelo Santos

Data: 28/02/2021

1) Com base nas tabelas abaixo, realize as consultas solicitadas utilizando o banco de dados PostgreSQL e depois realize as mesmas consultas utilizando o Framework Django e seus métodos de filtros e ordenação.

Pessoas			
id	**nome**	**Id_cargo**	**admissao**
1	Antonio Carlos	1	1999-05-10
2	Marcos Paulo	2	1998-04-05
3	Samanta Oliveira	2	2005-06-20
4	Beatriz Pires	1	2003-12-10
5	José dos Santos	3	1999-01-17
6	Maria das Graças	1	2007-06-07

Cargos	
id	**nome_cargo**
1	Gerente de Projetos
2	Tecnologia da Informação
3	Serviços Gerais
4	Departamento Pessoal

SQL : select p.nome, c.cargo from pessoas p join cargos c on p.id_cargo = c.id order by CONVERT (datetime, p.admissao) asc

A. Retorne por ordem crescente de admissão o nome dos funcionários e seus respectivos cargos na empresa Django : funcionarios = Pessoas.objects.all().values('nome', 'cargo__nome_cargo').order_by('admissao')

B. Considerando que a tabela Pessoas possua mais registros que o ilustrado acima, retorne o nome do funcionário e o nome do cargo do funcionário mais antigo na empresa.

SQL: select top 1 * from pessoas order by CONVERT(datetime, admissao) asc

Django: Pessoas.objects.all().earliest()

C. Retorne todos os nomes dos cargos com a quantidade de funcionários em cada um deles.

SQL : select c.nome_cargo, COUNT(p.nome) from pessoas p join cargos c on c.id = p.id_cargo

Django: Pessoas.objects.annotate(numero_cargos=Count('cargos'))

2) Enumere as colunas

```
(5) var e = function(){};
(1) var b = [];
(2) var c = '11';
(3) var a = {};
(4) var d = 5;
(6) var f = new function(){};
```

1 - Array

2 - String

3 - Object

4 - Number

5 - Function

6 - Instância/Classe

3) Qual o valor da variável *a*?

```
var a = 'Fulano';

function test(b){

        a = 'Beltrano';
}
test(a);
alert(a);
```

Responsta= Beltrano

4) Utilizando a biblioteca JQuery, escreva um código que esconda da tela o elemento com id "box" quando algum elemento da classe "btn" seja clicado.

Resposta= $('button').click(function(){ $("#box").hide()})

5) Criar relacionamentos 1-1, 1-N e N-N usando o django.

Resposta = 1-1

```
Class Post(models.Model):
  text = models.Charfield()
```

```
Class Pessoa(models.Model):
post = models.models.OneToOneField(Post, ...)
```

Resposta = 1-N

```
Class Post(models.Model):
  escritor = models.ForeignKey(Pessoa, ...)

Class Pessoa(models.Model):
  ...
```

Resposta = N-N

```
Class Post(models.Model):
  ...

Class Pessoa(models.Model):
  posts = models.ManyToManyField(Post, ...)
```

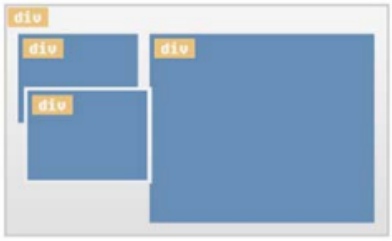
6) Considerando o código:


```
<!-- CSS -->
<style rel="stylesheet" type="text/css">
        .tudo { display: table; }
        .menu { float: left; width: 30%; }
        .conteudo { margin-left: 35%; }
</style>

<!-- HTML -->
<div class="tudo">
        <div class="menu">Menu</div>
        <div class="conteudo">Conteúdo</div>
</div>
```

Este código gera qual layout abaixo?

(A)



(B)

(C) X

7) Utilize Django e postgres para criar um crud de alvos com as seguintes características.

Campos: identificador, nome, latitude e longitude, data de expiração.
A tela inicial precisa apresentar mapa onde sera possível visualizar os alvos cadastrados e uma opção para incluir um novo alvo.

Na tela inicial ao clicar na opção incluir novo alvo abrir uma modal apresentando os campos em branco para preenchimento e opções para gravar e cancelar.

No mapa ao clicar sobre um alvo abrir a modal com os campos preenchidos e opções para salvar, cancelar e excluir.

8) Compartilhar o fonte do crud utilizando o GitHub.

https://github.com/johnatasr/Join-Challenge