Teste para candidatura Proffer

Time de Dados

Esta avaliação tem como intuito avaliar o entendimento sobre conhecimentos em SQL, conhecimento na linguagem Python, manipulação de dados, a utilização de APIs e a documentação de projetos

Levaremos em conta qualidade de código e erros ortográficos.

Instruções

Neste case, teremos três atividades. Na primeira etapa, você precisará desenvolver um banco de dados simples e criar as seguintes tabelas descritas abaixo (o banco de dados poderá ser local e apresentado durante a entrevista). Posteriormente, você deverá responder a algumas perguntas usando SQL. Lembre-se de gerar uma massa de dados falsos para popular as tabelas.

Os scripts de criação do banco de dados, schemas, tabelas, inserção e análises deverão ser salvos em um repositório git público, e o link deve ser disponibilizado. Lembre-se de documentar seu projeto no git da melhor forma possível, demonstrando sua lógica de resolução; esse cuidado também será levado em consideração. Você terá 2 dias para a resolução da atividade.

Se você aceitar participar do processo, terá o prazo de entrega de 3 dias (até quinta-feira), a partir da data de entrega deste teste.

Boa sorte!

Etapas

O desenvolvimento se dará em 2 etapas (leia o documento até o final), tente concluir o máximo de etapas que conseguir, mesmo que não consiga completar, levaremos em conta outros fatores.

1 - (Etapa Obrigatória) Prática em Bancos de Dados

O contexto de negócio das tabelas seguintes será, propositalmente, oposto aos produtos desenvolvidos na Proffer. O objetivo será testar a capacidade de adaptação em situações de regras de negócio fora do padrão esperado.

As tabelas abaixo se referem a dados de apostas esportivas . Crie e popule as seguintes tabelas:

Obs:Você poderá utilizar qualquer lógica/método da sua escolha para popular as tabelas, porém os dados devem seguir a lógica de relacionamento entre as tabelas.

Tabela **resultado**:
Data_acesso: o dia que o jogador realizou as ações.
Clientes_id: o id do cliente, essa coluna pode ser usada para buscar as informações da tabela **clientes**.
Buyin: o valor total apostado pelo jogador.
Winning: o valor total ganho pelo jogador, quando negativo, o prejuízo total do jogador.
Rake: o lucro da empresa com esse jogador.

Tabela **clientes**:
Id: o id do cliente, pode ser cruzado com a informação de *clientes_id* na tabela **resultado**.
Sexo: sexo do jogador, sendo m=mascluno e f=feminino.
Data_nascimento: ano, mês e dia de nascimento do jogador.
Data_cadastro: data e hora de quando o jogador realizou cadastro.
Cidade: cidade onde mora o jogador.
Sigla: UF onde mora o jogador.

Considerando essas tabelas, responda o código em SQL que responde às seguintes perguntas:

a) Quanto de rake foi gerado por cada Geração* de jogadores?
b) Qual foi o rake gerado por mês?
c) Qual sexo tem uma maior proporção de ganhadores**?

*Para essa atividade, considere cada geração tendo o seguinte critério:

- Veteranos, geração formada por pessoas que nasceram entre 1925 e 1940.
- Baby Boomers são os nascidos entre 1941 e 1959.
- Geração X, que compreende o período de 1960 a 1979.
- Geração Y é composta por indivíduos que nasceram entre 1980 e 1995.
- Geração Z é composta com os nascidos a partir de 1996 até 2010.
- Geração Alpha, engloba todos os nascidos a partir de 2010 até a presente data.

**Como ganhador, considere um jogador com Winning maior que 0

2 - (Etapa Opcional) Acessando APIs com Python

Essa etapa será opcional e não-eliminatória, o participante pode ficar à vontade em fazê-la ou não, porém caso escolha por executá-la, será avaliado o desempenho seguindo os mesmos padrões da etapa anterior.

Para essa atividade queremos que você utilize essa API. Nela você encontrará resultados de partidas de diversos esportes. Queremos que você utilize o endpoint de scores para buscar determinadas informações do sport soccer_brazil_campeonato na resposta da API e salvar numa tabela chamada partidas_brasileirao_serie_a_2023 contendo as seguintes colunas:

Obs: Lembre-se que o código utilizado também deverá ser versionado e documentado no mesmo repositório da etapa anterior.

- datahora_partida
- data_partida
- time_casa
- time_fora
- gols_time_casa
- gols_time_fora

Obs: trazer apenas os resultados do ano de 2023.

Essas informações devem ser salvas no banco de dados local desenvolvido, o mesmo utilizado para as outras atividades.