Atividade 1: Relacionamento Um-para-Um

Descrição:
Crie dois modelos no Django: **Usuário** e **Perfil**. O modelo **Perfil** deve conter os campos bio (texto) e data_nascimento (data), e deve estar relacionado ao modelo **Usuário** usando um relacionamento de um-para-um.

Tarefas:

1. Implemente os modelos no arquivo models.py.
2. Crie e aplique as migrações correspondentes no banco de dados.
3. No shell do Django (python manage.py shell), insira dados para um usuário e crie um perfil correspondente.

Atividade 2: Relacionamento Um-para-Muitos

Descrição:
Adicione um modelo **Postagem** que esteja relacionado ao modelo **Usuário** com um relacionamento de um-para-muitos. Cada postagem deve ter os campos titulo (texto) e conteudo (texto).

Tarefas:

1. Implemente os modelos no arquivo models.py, definindo a relação com um campo ForeignKey.
2. Crie e aplique as migrações no banco de dados.
3. Use o shell do Django para adicionar postagens a um usuário e liste todas as postagens de um usuário específico.

Atividade 3: Relacionamento Muitos-para-Muitos

Descrição:
Adicione um modelo **Grupo** e relacione-o ao modelo **Usuário** com um relacionamento muitos-para-muitos. Cada grupo deve ter os campos nome (texto) e descricao (texto).

Tarefas:

1. Implemente os modelos no arquivo models.py, utilizando um campo ManyToManyField.
2. Crie e aplique as migrações no banco de dados.
3. No shell do Django, adicione usuários a diferentes grupos e liste todos os usuários pertencentes a um grupo específico.

Atividade 4: Consultas Básicas com ORM

Descrição:
Utilizando o modelo **Postagem** criado anteriormente, escreva consultas no shell do Django para:

1. Filtrar todas as postagens de um usuário específico.
2. Obter uma postagem pelo ID.
3. Listar todas as postagens que contêm uma palavra-chave no título (use .filter() com icontains).

Atividade 5: Consultas Avançadas com ORM

Descrição:
Utilizando os modelos **Usuário**, **Postagem** e **Perfil**, escreva consultas no shell do Django para:

1. Obter todas as postagens de usuários que têm um perfil associado.
2. Reduzir o número de consultas usando .select_related() para carregar os dados de perfil de cada usuário junto com suas postagens.
3. Criar uma consulta agregada para contar o número total de postagens no sistema.

Atividade 6: Trabalhando com Relacionamentos

Descrição:

1. Adicione um campo categoria ao modelo **Postagem**. Este campo deve ser uma escolha entre opções como "Tecnologia", "Esportes", e "Arte".
2. Crie uma consulta para contar o número de postagens por categoria.
3. Liste todas as postagens de um grupo específico utilizando .prefetch_related() para otimizar as consultas de muitos-para-muitos.