Insper

Exercícios de revisão - Megadados

Um dos setores de tecnologia em franca expansão é o agrotech. Avanços em telecomunicações, computação e sistemas de informação viabilizam a aplicação de tecnologia da informação para o aumento da produtividade no campo.

Você foi contratado por uma empresa de automação agrícola. Sua tarefa é implementar um sistema para ajudar na aplicação de inseticidas no campo. Em conversas com o cliente, você levantou as seguintes informações sobre o domínio de negócios:

- Uma fazenda tem vários lotes. É importante dividir a fazenda em lotes para implementar a rotação das plantações: em um ano planta-se milho em um lote, noutro ano planta-se soja, etc.
- Cada lote tem apenas um tipo de plantação por vez (ou seja, não tem lote hibrido milho-soja)
- Um inseticida pode servir para vários tipos de plantação, mas não para todos. De modo similar, uma plantação pode receber diferentes inseticidas, mas provavelmente não todos.
- Para cada combinação de tipos de plantação e inseticida existe uma frequência mensal de aplicação.
- Cada inseticida tem um certo custo por hectare para ser aplicado.
- Cada lote vai receber um inseticida específico, determinado pelo engenheiro agrícola.

Questão 1: Liste as entidades deste problema e seus atributos, usando a notação de *schema* do modelo relacional.

Fazenda(id_fazenda, nome)

Lote(id_lote, id_fazenda, id_plantacao, id_inseticida, area)

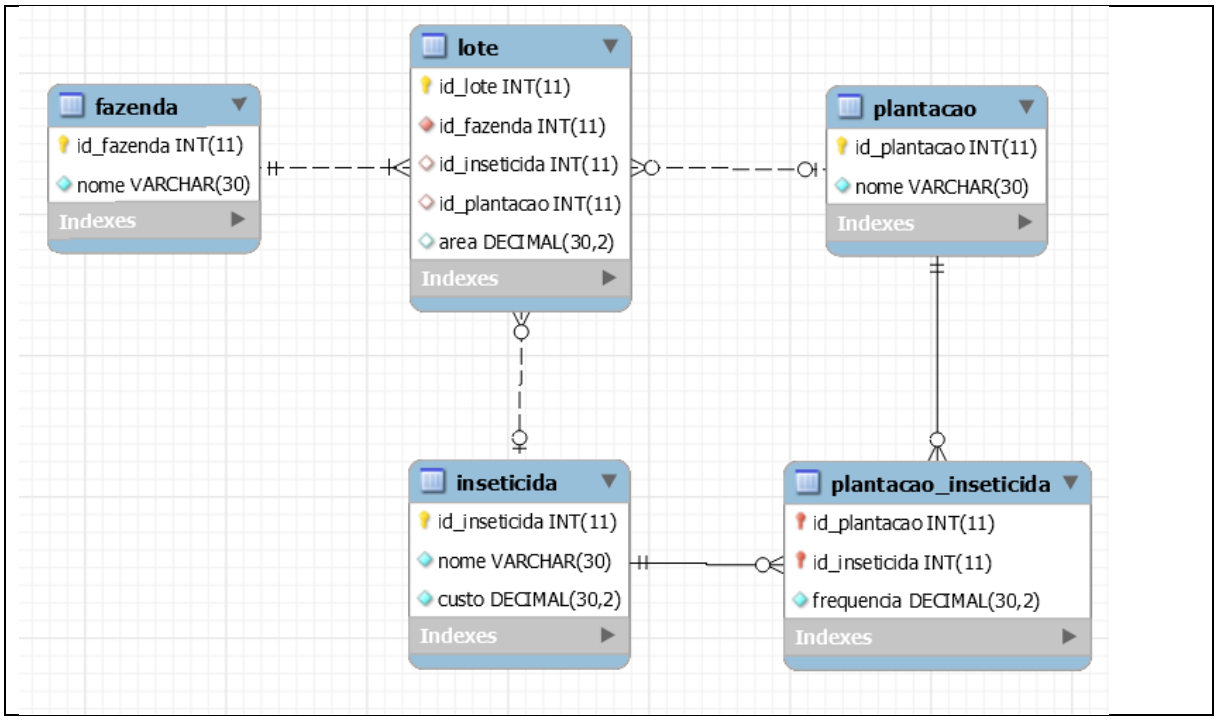
Plantação(id_plantacao, nome)

Inseticida(id_inseticida, custo_por_hectare)

Inseticida_Plantacao(id_plantacao, id_inseticida, frequencia)

Insper

Questão 2: Desenhe o diagrama do modelo relacional deste problema. Não esqueça de indicar claramente as chaves primárias, chaves estrangeiras, e a cardinalidade dos relacionamentos.



Questão 3: Escreva o script de criação da tabela referentes aos lotes

```
USE agro;

DROP TABLE IF EXISTS lote;
CREATE TABLE lote (
    id_lote INT NOT NULL AUTO_INCREMENT,
    id_fazenda INT NOT NULL,
    id_inseticida INT,
    id_plantacao INT,
    area DECIMAL(30 , 2 ),
    PRIMARY KEY (id_lote),
    FOREIGN KEY (id_fazenda)
        REFERENCES fazenda (id_fazenda),
    FOREIGN KEY (id_inseticida)
        REFERENCES inseticida (id_inseticida),
    FOREIGN KEY (id_plantacao)
        REFERENCES plantacao (id_plantacao)
);
```

Insper

Questão 4: Escreva a DML para as seguintes tarefas:

a. Crie 3 lotes para a fazenda: um de 4 hectares, outro de 3 hectares, e outro de 5 hectares.

```
INSERT INTO lote (id_fazenda, id_inseticida, id_plantacao,
area) VALUES (1, 1, 1, 3), (1, 1, 1, 4), (1, 1, 1, 5);
```

b. Corrija um erro no item anterior: mude o tamanho do lote de 3 hectares para 3,5 hectares.

```
UPDATE lote SET area = 3.5 WHERE id_lote = 1;
```

c. Como obter o id do lote em uma aplicação real? É realista pedir ao usuário que decore o id de cada lote?

Certamente não é realista pedir que os usuários memorizem os ids de cada lote. A aplicação deve ter uma interface de usuário que permita a seleção do lote de interesse de modo simples.

Questão 5: Escreva queries para as seguintes tarefas:

a. Qual a maior fazenda?

```
SELECT
    nome, SUM(area) AS total
FROM
    fazenda INNER JOIN lote USING (id_fazenda)
ORDER BY total DESC
LIMIT 1;
```

b. Quais as plantações com área total maior que 10 hectares?

```
SELECT
    plantacao.nome, SUM(area) AS total
FROM
    lote,
    plantacao
WHERE
    lote.id_plantacao = plantacao.id_plantacao
GROUP BY lote.id_plantacao
HAVING total > 10;
```

c. Qual o custo total atual com inseticidas?

```
SELECT
    SUM(lote.area *
        plantacao_inseticida.frequencia *
        inseticida.custo)
FROM
    lote,
    inseticida,
    plantacao_inseticida
WHERE
    lote.id_inseticida = plantacao_inseticida.id_inseticida
    AND lote.id_plantacao = plantacao_inseticida.id_plantacao
    AND lote.id_inseticida = inseticida.id_inseticida;
```

d. Quais inseticidas não foram usados?

```
SELECT
    inseticida.nome
FROM
    inseticida
        LEFT OUTER JOIN
    lote USING (id_inseticida)
WHERE
    id_lote IS NULL;
```

e. Qual o menor custo possível com inseticidas?

```
SELECT
  SUM(lote.area * tab_min.custo_mensal)
FROM
  lote
    INNER JOIN
  ( SELECT
      id_plantacao,
      MIN(plantacao_inseticida.frequencia * inseticida.custo)
AS custo_mensal
    FROM
      inseticida
    INNER JOIN plantacao_inseticida USING (id_inseticida)
    GROUP BY id_plantacao ) tab_min
  USING (id_plantacao);
```

Ok, essa está meio complicada. Como chegar nesse resultado?
Em primeiro lugar vamos pensar o seguinte: o menor custo possível com inseticidas acontece quando aplicamos o inseticida mais barato em cada lote. Qual o custo do inseticida em um lote?

$$Custo\ do\ inseticida = (Área\ do\ lote) \times (frequência\ de\ aplicação) \times (custo\ por\ aplicação)$$

Não temos como mudar a área do lote nem o tipo de plantação feito, logo o que estamos buscando é minimizar o produto (frequência de aplicação) vezes (custo por aplicação) para cada tipo de plantação. Por exemplo, suponha que tenhamos os seguintes dados:

Tabela plantação_inseticida

id_plantação	id_inseticida	Frequência
5	1	4
5	2	5
5	3	4
7	2	2
7	3	2

Tabela inseticida

id_inseticida	nome	Custo
1	Lindinho	110
2	Docinho	90
3	Fofinho	100

Se fizermos um join destas tabelas, temos o seguinte:

```
SELECT * FROM inseticida INNER JOIN plantacao_inseticida
USING(id_inseticida)
```

id_plantacao	id_inseticida	frequencia	nome	custo
5	1	4	Lindinho	110
5	2	5	Docinho	100
5	3	4	Fofinho	100
7	2	2	Docinho	90
7	3	2	Fofinho	100

Estamos interessados apenas no produto frequência vezes custo, e no id_plantação (precisamos dele para juntar com a tabela dos lotes depois):

```
SELECT id_plantacao, frequência * custo AS custo_mensal
FROM inseticida INNER JOIN plantacao_inseticida
USING(id_inseticida)
```

Id_plantacao	custo_mensal
5	440
5	450
5	400
7	180
7	200

Podemos agora agrupar esses resultados por tipo de plantação e reter apenas o menor custo mensal!

```
SELECT id_plantacao, MIN(frequência * custo) AS custo_mensal
FROM inseticida INNER JOIN plantacao_inseticida
USING(id_inseticida)
GROUP BY id_plantacao
```

Id_plantacao	custo_mensal
5	400
7	180

Valores mínimos de custo mensal por hectare para cada id_plantacao

Vamos chamar essa tabela de tab_min. Agora voltemos à tabela de lotes: vamos uni-la com tab_min. Suponha que a tabela lote tem o seguinte conteúdo:

id_lote	id_fazenda	id_inseticida	id_plantacao	Área
1	55	7	5	10
2	66	8	7	15
3	55	8	5	10
4	55	2	5	5
5	44	1	5	20
6	44	5	7	10

Fazendo o join da tabela lote com a tabela tab_min:

```
SELECT * FROM lote INNER JOIN tab_min USING (id_plantacao);
```

id_lote	id_fazenda	id_inseticida	id_plantacao	Área	custo_mensal
1	55	7	5	10	400
2	66	8	7	15	180
3	55	8	5	10	400
4	55	2	5	5	400
5	44	1	5	20	400
6	44	5	7	10	180

A quantidade de interesse é o produto da área pelo custo mensal mínimo por hectare (coluna custo_mensal), o resto pode jogar fora:

```
SELECT area*custo_mensal FROM lote INNER JOIN tab_min USING
(id_plantacao);
```

area * custo_mensal
4000
2700
4000
2000
8000
1800

Por fim, estamos na verdade interessados na soma destes custos:

```
SELECT SUM(area*custo_mensal)
FROM lote INNER JOIN tabmin USING (id_plantacao);
```

SUM(area * custo_mensal)
22500

Esse é o resultado final que estávamos procurando! Juntando todos os pedaços de query que desenvolvemos temos a solução completa:

```
SELECT
  SUM(lote.area * tab_min.custo_mensal)
FROM
  lote
    INNER JOIN
  ( SELECT
      id_plantacao,
      MIN(plantacao_inseticida.frequencia * inseticida.custo)
        AS custo_mensal
    FROM
      inseticida
      INNER JOIN plantacao_inseticida USING (id_inseticida)
    GROUP BY id_plantacao ) tab_min
  USING (id_plantacao);
```

f. Escreva a *query* mais cruel que você consegue imaginar! 😊