

Iniciado em quinta, 19 nov 2020, 19:05

Estado Finalizada

Concluída em quinta, 19 nov 2020, 19:10

Tempo empregado 5 minutos 4 segundos

Avaliar 10,00 de um máximo de 10,00(100%)

Questão 1

Completo

Atingiu 2,00 de 2,00

Considere as seguintes afirmações comparativas entre Pandas, Spark RDD e Spark SQL.

- I. É possível responder à consulta “Qual é a média dos salários recebidos por nível do cargo e por sexo no ano de 2019?” definida sobre a constelação de fatos da BI Solutions utilizando Pandas, especificando a consulta SQL textual como um parâmetro do método `spark.sql()` e utilizando métodos do módulo `pyspark.sql`, obtendo, em todos os casos, a mesma resposta.
- II. Considerando o modelo de 3Vs que define o conceito de big data, pode-se afirmar que, utilizando Apache Spark é possível manipular um gigantesco volume de dados, oferecer suporte para dados estruturados apenas e prover processamento de dados em lote e streaming.
- III. Tanto a biblioteca Pandas quanto o módulo `pyspark.sql` utilizam o conceito de DataFrames, sendo que a diferença está relacionada ao fato de que os DataFrames utilizados pelo módulo `pyspark.sql` são baseados em RDDs e, portanto, embutem aspectos do processamento paralelo e distribuído.
- IV. Especificar uma consulta usando a linguagem SQL como um parâmetro do método `spark.sql()` indica que o usuário deve especificar quais dados devem ser obtidos ao invés de como esses dados devem ser obtidos, sendo que a forma como os dados são obtidos é definida pelo otimizador de consultas de forma transparente.
- V. Especificar uma consulta usando Pandas ou os métodos do módulo `pyspark.sql` indica que o usuário deve especificar quais dados devem ser obtidos ao invés de como esses dados devem ser obtidos, sendo que a forma como os dados são obtidos é definida pelo otimizador de consultas de forma transparente.

Marque:

Escolha uma opção:

- ☒ a. Se as afirmativas I, III e IV estão corretas.
- ☐ b. Se as afirmativas II, III e V estão corretas.
- ☐ c. Se as afirmativas I, III e V estão corretas.
- ☐ d. Se as afirmativas I, II, III e V estão corretas.
- ☐ e. Se as afirmativas I, II, III e IV estão corretas.

Considerando a constelação de fatos da BI Solutions, assinale a alternativa que corresponde à resolução da seguinte consulta usando métodos do módulo `pyspark.sql`: "Liste, para cada funcionário, sua chave primária, seu nome e a soma dos salários recebidos. Ordene o resultado final em ordem ascendente pela chave primária do funcionário."

Escolha uma opção:

- ☐ a.

```
pagamento\
join(funcionario, on="funcPK")\
.select("funcPK", "funcNome", .sum("salario"))\
.groupBy("funcPK", "funcNome")\
.orderBy("funcPK")\
.show()
```
- ☐ b.

```
pagamento\
join(funcionario, on="funcPK")\
.select("funcPK", "funcNome")\
.groupBy("funcPK", "funcNome")\
.sum("salario")\
.orderBy("funcPK")\
.show()
```
- ☒ c.

```
pagamento\
join(funcionario, on="funcPK")\
.select("funcPK", "funcNome", "salario")\
.groupBy("funcPK", "funcNome")\
.sum("salario")\
.orderBy("funcPK")\
.show()
```
- ☐ d.

```
pagamento\
join(funcionario, on="funcPK")\
.select("funcPK", "funcNome", "salario")\
.groupBy("funcPK", "funcNome")\
.sum("salario")\
.orderBy("funcPK", ascending=False)\
.show()
```
- ☐ e.

```
pagamento\
join(funcionario, on="funcPK")\
.select("funcPK", "funcNome")\
.groupBy("funcPK", "funcNome")\
.sum("salario")\
.orderBy("funcPK")\
.show()
```

Considerando a constelação de fatos da BI Solutions, assinale a alternativa que **NÃO** corresponde à resolução da seguinte consulta usando métodos do módulo pyspark.sql: "Liste o ano e o valor do salário recebido pelo funcionário identificado por funcPK = 147 na data identificada por dataPK = 5. "

Escolha uma opção:

- ☐ a. pagamento\
join(data, on="dataPK")\
.filter("funcPK = 147")\
.filter("dataPK = 5")\
.select("dataAno", "salario")\
.show()
- ☒ b. pag = pagamento.filter("funcPK = 147")
dat = data.filter("dataPK = 5")
pag.join(dat, on="dataPK").show()
- ☐ c. pag = pagamento.filter("funcPK = 147").select("dataPK", "salario")
dat = data.filter("dataPK = 5").select("dataPK", "dataAno")
pag.join(dat, on="dataPK").select("dataAno", "salario").show()
- ☐ d. pagamento\
join(data, on="dataPK")\
.select("dataAno", "salario")\
.filter("dataPK = 5 AND funcPK = 147")\
.show()
- ☐ e. pagamento\
join(data, on="dataPK")\
.filter("funcPK = 147")\
.select("dataAno", "salario")\
.filter("dataPK = 5")\
.show()

Considere as seguintes afirmações sobre os métodos do módulo `pyspark.sql`.

- I. Ambos os métodos `join()` e `crossJoin()` unem dois `DataFrames`, porém no método `join()` as linhas são combinadas com base na integridade referencial existente entre essas linhas, enquanto que o método `crossJoin()` combina quaisquer linhas, independente da integridade referencial existente entre elas.
- II. O método `join()` pode realizar diferentes tipos de junção, de acordo com o parâmetro `how`, sendo o valor padrão `inner`, mas também podendo definir outros valores como `cross`, `outer`, `full`, `left`, `right`.
- III. Ambos os métodos `union()` e `unionAll()` não eliminam as linhas repetidas do resultado final, sendo necessário aplicar o método `distinct()` na sequência para que as linhas repetidas sejam eliminadas.
- IV. Ambos os métodos `rollup()` e `cube()` criam vários níveis de agregação usando como base as colunas especificadas como parâmetro, gerando o mesmo resultado.
- V. O método `orderBy()` ordena os dados exibidos como resposta à consulta de modo descendente por padrão, sendo necessário definir explicitamente o modo crescente caso seja necessário.

Indique se cada uma das afirmações é V (Verdadeira) ou F (Falsa).

Escolha uma opção:

- ☒ a. V - V - V - F - F
- ☐ b. V - F - V - V - F
- ☐ c. V - F - F - V - V
- ☐ d. V - V - V - V - V
- ☐ e. F - V - F - F - V

Considere a constelação de fatos da BI Solutions e um DataFrame que armazena todos os dados da relação data. Considere o esquema deste DataFrame ilustrado na Figura 1.

Considere a sequência de comandos ilustrada na Figura 2.

Assinale a alternativa que corresponde ao esquema do DataFrame data após a aplicação dos comandos ilustrados na Figura 2.

```
root
|-- dataPk: integer (nullable = true)
|-- dataCompleta: string (nullable = true)
|-- dataDia: integer (nullable = true)
|-- dataMes: integer (nullable = true)
|-- dataBimestre: integer (nullable = true)
|-- dataTrimestre: integer (nullable = true)
|-- dataSemestre: integer (nullable = true)
|-- dataAno: integer (nullable = true)
```

Figura 1: Esquema do DataFrame data.

```
data\
.withColumnRenamed("dataPK", "chave primária de data")\
.withColumnRenamed("dataPK", "chave primária")\
.withColumn("diferenca", data["dataAno"] - data["dataDia"])\
.withColumnRenamed ("diferenca", "diferença entre ano e dia")\
.printSchema()
```

Figura 2: Sequência de comandos sobre o DataFrame data.

Escolha uma opção:

☐ a.

```
root
|-- chave primária de data: integer (nullable = true)
|-- dataCompleta: string (nullable = true)
|-- dataDia: integer (nullable = true)
|-- dataMes: integer (nullable = true)
|-- dataBimestre: integer (nullable = true)
|-- dataTrimestre: integer (nullable = true)
|-- dataSemestre: integer (nullable = true)
|-- dataAno: integer (nullable = true)
|-- diferenca: integer (nullable = true)
|-- diferença entre ano e dia: integer (nullable = true)
```

☐ b.

```
root
|-- chave primária de data: integer (nullable = true)
|-- dataCompleta: string (nullable = true)
|-- dataDia: integer (nullable = true)
|-- dataMes: integer (nullable = true)
|-- dataBimestre: integer (nullable = true)
|-- dataTrimestre: integer (nullable = true)
|-- dataSemestre: integer (nullable = true)
|-- dataAno: integer (nullable = true)
|-- chave primária: integer (nullable = true)
|-- diferenca: integer (nullable = true)
|-- diferença entre ano e dia: integer (nullable = true)
```

☐ c.

```
root
|-- chave primária: integer (nullable = true)
|-- dataCompleta: string (nullable = true)
|-- dataDia: integer (nullable = true)
|-- dataMes: integer (nullable = true)
|-- dataBimestre: integer (nullable = true)
|-- dataTrimestre: integer (nullable = true)
|-- dataSemestre: integer (nullable = true)
|-- dataAno: integer (nullable = true)
|-- diferenca: integer (nullable = true)
|-- diferença entre ano e dia: integer (nullable = true)
```

☒ d.

```
root
|-- chave primária de data: integer (nullable = true)
|-- dataCompleta: string (nullable = true)
|-- dataDia: integer (nullable = true)
|-- dataMes: integer (nullable = true)
|-- dataBimestre: integer (nullable = true)
|-- dataTrimestre: integer (nullable = true)
|-- dataSemestre: integer (nullable = true)
|-- dataAno: integer (nullable = true)
|-- diferença entre ano e dia: integer (nullable = true)
```

☐ e.

```
root
|-- chave primária: integer (nullable = true)
|-- dataCompleta: string (nullable = true)
|-- dataDia: integer (nullable = true)
|-- dataMes: integer (nullable = true)
|-- dataBimestre: integer (nullable = true)
|-- dataTrimestre: integer (nullable = true)
|-- dataSemestre: integer (nullable = true)
|-- dataAno: integer (nullable = true)
|-- diferença entre ano e dia: integer (nullable = true)
```

[◀ Exercícios - soluções](#)

Seguir para...



[Envio da Avaliação Final da disciplina ▶](#)