CENTRO DE ENSINO SUPERIOR E DESENVOLVIMENTO – CESED

CENTRO UNIVERSITÁRIO – UNIFACISA

CURSO: SISTEMAS DE INFORMAÇÃO

PROFESSOR: JOSÉ ANDERSON RODRIGUES DE SOUZA

COMPONENTE CURRICULAR: CONECTAR BANCO DE DADOS COM P-O-O

LISTA DE EXERCÍCIO 01

(Persistência a Dados, Tipos de Dados, XML, JSON e JDBC)

1. O que você entende por Persistência de Dados?

2. Quais as diferenças entre objetos transientes e objetos persistentes? Explique.

3. Qual a principal diferença entre Banco de Dados Relacional e Banco de Dados Orientada a Objetos?

4. Qual o objetivo do mapeamento objeto-relacional (ORM)?

5. Defina:

a) Dados estruturados;

b) Dados semiestruturados;

c) Dados não estruturados.

6. Qual o nome da biblioteca responsável pela extração/captura de dados disponíveis em arquivos HTML ou XML? Explique.

7. Os arquivos do tipo XML (eXtensible Markup Language) surgiram como forma de estruturação e troca de dados pela internet. Dentre suas principais característica preencha os seguintes questionamentos:

a) Sintaxe inicial na primeira linha do arquivo.xml

b) Os dados são organizados em formato hierárquico ou tabular?

c) Quais são as formas de representação de um documento XML.

Justique.

8. Elabore um documento xml sobre produtos disponíveis para venda em empresas do comércio eletrônico/móveis/imóveis/roupas, a partir das seguintes condições:

▪ O produto deve possuir 5 características;

▪ Cada produto deve ter um nome de identificação;

▪ No documento deverá ter pelo menos dois produtos preenchidos.

9. Defina o que é um documento JSON e quais suas principais características.

10.O que significa o processo de serialização (JSON.stringify) e desserialização (JSON.parse) de documentos do tipo JSON?

11. Faça um exemplo de documento JSON a partir de dados sobre serviços de vendas online.

• Utilize dados do tipo, string, inteiro, array e objetos.

12. Quais são as principais diferenças entre documentos do tipo JSON e XML.

13. Para que serve utilizar JDBC com Sistemas de Gerenciamento de Banco de Dados.

14. Quais são os principais componentes durante a implementação do JDBC? Explique.

15. Cites restrições sobre a utilização do JDBC para sistemas atuais.

RESPOSTAS

1 –

Persistência de dados é um conceito muito importante no campo da programação, ele refere-se a capacidade e a necessidade de ser armazenar dados em um determinado banco, independe do mesmo ser relacional ou não. A grande maioria das aplicações de hoje necessitam armazenar dados de clientes, produtos, api's externas e coisas do gênero, tudo isso está relacionado com o conceito de persistência.

Isto é, uma abstração a respeito da persistência de dados é entender a necessidade de armazenamento de dados de forma relacional ou em coleções. Porém não é apenas isso, geralmente esses dados veem na forma de objetos então são utilizadas diferentes estratégias para fazer o armazenamento adequado. Em outras palavras persistir pode ser a entendido como uma forma de salvar um objeto logo após a execução e encerramento de uma aplicação.

2 –

Existem dois tipos de objetos relacionados a programação. O primeiro destes trata-se dos objetos transientes que são aqueles objetos que uma vez que a aplicação deixe de funcionar

ele é perdido, portanto só existem enquanto o programa está em execução. Por outro lado, os objetos persistentes são aqueles que independente da aplicação, permanecerão salvos no banco de dados.

Obviamente ambos possuem aplicações diferentes no contexto da programação, por exemplo, um software em desenvolvimento pode renunciar a um banco de dados e utilizar os objetos transientes para testar uma interface web, por exemplo. Por outro lado, grande parte das aplicações necessitam de bancos de dados para salvar objetos pelos mais variados tipos de necessidades, desde um simples cadastro de clientes a uma API de dados climáticos que poderão ser processados para obter uma previsão do tempo.

Portanto, quando falamos de persistência de dados estamos justamente trabalhando com a necessidade de armazenar esses dados em um banco, para isso podemos usar diferentes estratégias para importar esses dados seja utilizando models para criar um banco de dados relacional ou notação JSON.

3 –

Diferentes aplicações usam diferentes tipos de banco de dados. Hoje a grande maioria do mercado de TI trabalha com os bancos de dados relacionais SQL, isto porque esse tipo de banco demonstrou-se muito eficaz para salvar os dados, mas não apenas isso, possui diferentes mecanismos de controle que garantem a integridade destes dados no banco e, por fim, uma linguagem fácil para obter esses dados quando necessário.

O fato é que um banco de dados relacional nem sempre é a melhor alternativa, sua implementação inicial não é simples, pelo contrário, quando se trata dos conceitos e a aplicabilidade das normalizações uma infinidade de tabelas podem surgir apenas para criar os relacionamentos adequados.

Para o caso das aplicações OO foram criadas alternativas ao relacional, isto porque os objetos geralmente possuem estado (valor) e comportamento (operações), podem ter uma estrutura complexa e operações específicas definidas pelo desenvolvedor. Salvar esse tipo de dados em um banco relacional seria muito complexo. Por isso os bancos de dados OO possuem a capacidade de estender a existência dos objetos armazenando-os permanentemente no banco de dados.

4 –

Salvar um objeto no banco de dados pode ser uma tarefa complexa. A solução adotada para resolver essa dificuldade trata-se do mapeamento objeto-relacional. É uma técnica de desenvolvimento capaz de representar um objeto de maneira relacional no banco de dados. Uma ORM possui três componentes: o Modelo Orientado a Objetos, que representa os dados na aplicação como objetos; a Persistência Física, que corresponde ao modelo relacional em que os dados serão armazenados no banco de dados; e a Camada de Abstração, que atua como uma ponte para 'construir' e 'descontruir' objetos no modelo orientado a objetos para a persistência física no modelo relacional.

5 –

a) Os dados estruturados seguem, uma estrutura de organização com formato predefinido, segue o modelo de linhas e colunas. Para cada coluna temos um atributo específico e cada linha uma nova entrada na tabela. Suas principais características remetem ao formato tabular, a linguagem de consulta SQL e a facilidade no armazenamento, pesquisa e análise dos dados.
b) Os dados semiestruturados possuem algum tipo de estrutura, mas não chegam a seguir o formato completamente tabular. Na verdade, usam técnicas de encapsulamento para manter uma organização hierárquica no formato chave-valor. Por exemplo XML, JSOM, HTML e lembra também o REACT. Suas principais características incluem a hierarquia dos dados, a partir do aninhamento ou encapsulamento; uso de chave-valor; flexibilidade para adicionar novas chaves sem impactar a estrutura global.
c) Os dados não estruturados não possuem formato nem estrutura definida. Podem ser de natureza textual, multimídia ou qualquer outra forma que não consiga se encaixar em tabelas ou hierárquicas. Exemplos disso são textos livres, áudios e vídeos. Suas principais características são a falta de um formato predefinido; requer técnicas avançadas de processamento e linguagem natural e aprendizado de máquina para análise; contém informações valiosas, mas de difícil uso.

6 –

Em se tratando de bibliotecas para análise de arquivos XML e HTML temos duas alternativas interessantes. No caso do HTML podemos usar JSOUP que é uma biblioteca de código aberto para análise e manipulação de HTML com JAVA. Essa biblioteca é comumente utilizada no processo de WEB SCRAPING que se trata da raspagem de dados da WEB. Ele é

capaz de encontrar elementos HTML por meio de seletores CSS, extrair textos e atributos, além de modificar elementos HTML. No caso do Python temos uma importantíssima biblioteca chamada Beatifull Soup que além de da análise do HTML também consegue fazer o mesmo com XML.

Mais especificamente voltado ao XML podemos utilizar um módulo da biblioteca padrão do Python que se chama ElementTree cujo import é o xml.etree.ElementTree. Esse módulo fornece uma forma fácil e rápida de análise de XML, criar XMLs e pode manipular a estrutura de XMLs existentes.

7 –

a) <?xml version="1.0" encoding="UTF-8"?>
b) Hierárquico a partir da abertura e fechamento de tags de início e fim respectivamente, representadas pelos sinais <...> e </...> que são os delimitadores dos nomes das tags.
c) Existem duas formas de representar um arquivo XML. A representação textual que á forma mais básica de representação onde o arquivo é armazenado ou transmitido como um texto simples que facilite a leitura ou com a extensão .xml com sua estrutura hierárquica de abertura e fechamento de tags para representar dados. A segunda forma é a estrutura de árvore que surge a partir do elemento raiz que seria o principal tag na árvore de hierarquias do documento que encapsularia todo o resto.

8 –

produto.xml > ...

```xml
<?xml version="1.0" encoding="UTF-8"?>
<hardware>
    <produto>
        <placa_mae id="001">
            <nome>ASUS ROG Strix Z590-E</nome>
            <fabricante>ASUS</fabricante>
            <soquete>CPU LGA 1200</soquete>
            <chipset>Intel Z590</chipset>
            <memoria_suportada>DDR4 até 128GB</memoria_suportada>
            <slots_pci_express>3 x PCIe 4.0 x16</slots_pci_express>
        </placa_mae>
    </produto>
    <produto>
        <placa_mae id="002">
            <nome>Gigabyte B450 AORUS PRO</nome>
            <fabricante>Gigabyte</fabricante>
            <soquete>CPU AM4</soquete>
            <chipset>AMD B450</chipset>
            <memoria_suportada>DDR4 até 64GB</memoria_suportada>
            <slots_pci_express>1 x PCIe 3.0 x16, 1 x PCIe 2.0 x16</slots_pci_express>
        </placa_mae>
    </produto>
    <produto>
        <placa_mae id="003">
            <nome>MSI MPG B550 Gaming Edge WiFi</nome>
            <fabricante>MSI</fabricante>
            <soquete>CPU AM4</soquete>
            <chipset>AMD B550</chipset>
            <memoria_suportada>DDR4 até 128GB</memoria_suportada>
            <slots_pci_express>1 x PCIe 4.0 x16, 1 x PCIe 3.0 x16</slots_pci_express>
        </placa_mae>
    </produto>
</hardware>
```

9 –

JSON é acrônimo de JavaScript Object Notation, em outras palavras é uma linguagem de representação de um objeto Javascript. Trata-se do formato mais aberto e popular para representação e troca de dados. Sua leitura e manipulação tem um formato fácil para leitura humana e simples para os computadores processarem.

Trata-se de um conjunto não-ordenado de dados armazenados em um par chave-valor que inicia e termina com {}. Todos os nomes-chave são englobados em aspas duplas e o separador é o sinal ":". Ao fim de um dado usamos vírgulas para separar. Esse tipo de documento possui suporte a diversos tipos de valores como strings, inteiros, floats, booleans, nulos, arrays e objetos. Por outro lado não aceita comentários.

10 –

Serialização é o processo de capturar uma estrutura de dados, como um objeto JS, para que o mesmo possa ser armazenado, transmitido ou posteriormente reconstruído. Seu objetivo é converter a estrutura de dados em um formato que seja facilmente armazenado ou transportado. Esse processo é muito útil quando existe a necessidade de armazenar em um banco ou transmitir dados, fazemos a serialização para que ele tenha um formato que possa ser facilmente gravado ou enviado pela rede.

A maioria das linguagens de programação possuem métodos ou bibliotecas para realizar a serialização de objetos ou estruturas de dados. No caso do JS o método JSON.stringify é usado para realizar a serialização de um objeto JavaScript em uma representação JSON.

Na contramão temos a deserialização que é o processo inverso, isto é, depois que uma estrutura de dados é serializada ela pode ser deserializada para voltar para a estrutura original, isto é útil para recuperar dados armazenados ou transmitidos para que sejam usados em uma aplicação.

Em resumo o processo de serialização é transformar uma estrutura de dados em um JSON e a descerialização é reconstruir o JSON em uma estrutura de dados.

11 –

12 –

Tanto JSON quanto XML são arquivos com a capacidade de armazenas dados com a finalidade de transmitir ou mesmo armazenar em um banco. Dentre as principais diferenças existentes podemos notar a sintaxe que é bem diferente. No caso do XML cada dado é encapsulado entre tags de abertura e fechamento e tudo isso encapsulado em um elemento raiz; por outro lado o JSON possui uma estrutura diferente, o conteúdo fica disposto entre chaves e utiliza-se o princípio chave-valor para armazenar os dados. Um arquivo XML precisa iniciar uma declaração sobre o tipo de arquivo, o JSON baste iniciarmos com a abertura de chaves. O XML geralmente é mais utilizado para comunicação entre sistemas nas camadas superiores com os serviços web SOAP, enquanto o JSON é mais utilizado para consumo em APIs RESTful ou bancos de dados.

13 –

JDBC trata-se de uma interface baseada em Java para acesso de banco de dados através de SQL, a partir dele é possível acessar diretamente o banco.

Em outras palavras o JDBC fornece uma API para acessar bancos de dados a partir de aplicativos Java. Para que seja possível a conexão correta ele necessita do carregamento de drivers específicos do banco que atuam como ponte entre o Java e o SGBD. A conexão é feita

através do método getConnection da classe DriverManager e a execução dos códigos sql são feitos através dos Statements. O resultado pode ser obtido através do método ResultSet.

14 –

Os principais componentes de uma implementação JDBC são:

1- Aplicação Java, programa que deseja interagir com o banco de dados;
2- Drivers do banco, componentes específicos do banco que atuam como ponte entre o banco e a aplicação;
3- Banco de dados, local onde são armazenados os dados

O JDBC possui classes específicas para a conexão adequada com o banco de dados. Suas principais classes são:

- DriverManager com o método getConnection que aceita uma URL e retorna um objeto Connection.

- Connection que representa uma conexão ativa com o banco de dados. Ela é usada para criar objetos Statement.

- Statement que é usado para criar e executar consultadas SQL, ele é gerado a partir do método createStatement no objeto Connection

- ResultSet com seu método executeQuery que é usada para armazenas os dados para recuperação pela aplicação java.

15 –

Compatibilidade com drives: É preciso se certificar que o driver usado é compatível com o banco de dados que está em uso;

Segurança: O JDBC não fornece proteção contra ataques, como injeção de sql;

Gerenciamento de conexão: O JDBC não gerencia automaticamente o pool de conexões.

Desempenho: O uso inadequado pode gerar problemas de desempenho como consultas ineficientes, muitas consultas individuais;

Complexidade: Muito verboso, de difícil manutenção.