



Projeto Final

Entrega: 13/11, seguida de apresentação

Esta data não poderá ser prorrogada em hipótese alguma, portanto programem-se.

→ **Projeto com 2 alunos no máximo**

→ **Qualquer linguagem de programação não Web**

→ **Entregar no Tidia (em um único zip):**

+um txt com os nomes dos integrantes do grupo e, caso necessário, relatório com esclarecimentos necessários à correção;
+o projeto pronto para compilação.

→ **Apresentação presencial:** os trabalhos serão apresentados pelo grupo (presença obrigatória do grupo todo) durante o horário de aula. Os membros do grupo apresentarão o sistema funcionando, o código da aplicação, executarão exemplos demonstrativos, e responderão perguntas.

1. (3.0) Escreva um programa que, dada uma conexão Oracle, escreva documentos BSON a partir de cada tupla para inserção em uma base de dados MongoDB. Escreva seu programa de modo que a escrita de documentos BSON ocorra a partir de tabelas ou de resultados de consultas SQL quaisquer.

Execute o programa sobre toda a base de dados de futebol (atenção aos relacionamentos N-N – faça o mapeamento apenas para o caso de linking, de maneira cruzada); crie um script com seus comandos em um arquivo texto mongo.mql com os comandos separados por linha.

Para este exercício:

→ apenas no caso do relacionamento estado-cidade, faça o embedding dos estados nas cidades sempre;

→ coloque uma opção que permita indicar se os BSONs serão criados com embedding ou com linking;

→ em caso de linking, atenção à ordem de criação, e aos dados necessários antes de se fazer insert → use as chaves primárias do Oracle como `_id`;

→ crie uma coleção para cada tabela da base; para as consultas crie coleções denominadas consulta1, consulta2, etc

→ verifique os tipos BSON e faça o mapeamento BSON-Oracle correto dentro do seu código

<https://docs.mongodb.com/manual/reference/bson-types/>

→ valores null devem ser ignorados ou pode-se usar o tipo null do MongoDB.

2. (1.5) Em seu programa, escreva um procedimento que gere um script para a criação de índices MongoDB para as chaves secundárias da sua base a partir do dicionário de dados do Oracle; considere índices de um único atributo e índices compostos.

3. (1.5) A partir das restrições de integridade (check, not null) escreva manualmente um documento de validação em formato de script para toda sua base de dados.

→ <https://docs.mongodb.com/manual/core/document-validation/>

4. (1.5) Crie uma funcionalidade que, dada uma consulta SQL, escreva o correspondente comando find. Considere apenas consultas simples, que possuam uma lista de atributos, uma única tabela, e predicados simples com operadores AND apenas, ou OR apenas. Considere todas os operadores de comparação do MongoDB:

<https://www.mongodb.com/docs/manual/reference/operator/query-comparison/>

5. (1.0) Crie uma funcionalidade que, dada uma consulta SQL com junção, escreva o correspondente comando aggregate. Considere apenas junções simples, com duas tabelas e predicado de junção contendo apenas um atributo, isto apenas chaves referenciadas não compostas.

6. (1.5) Crie uma funcionalidade que, dada uma consulta SQL com comando de agregação, mas sem cláusula de GROUP BY, escreva o correspondente comando aggregate. Ofereça opção para 6 funções diferentes. Considere as funções de agregação do Oracle <https://docs.oracle.com/en/database/oracle/oracle-database/19/sqlrf/Aggregate-Functions.html#GUID-62BE676B-AF18-4E63-BD14-25206FEA0848> e os operadores de agregação do

<https://www.mongodb.com/docs/manual/reference/operator/aggregation/group/>.