Universidade Federal de Itajubá
Instituto de Matemática e Computação
COM-231 – BANCO DE DADOS II
Profª Vanessa Souza

BANCO DE DADOS II
DEFINIÇÃO DO TRABALHO PRÁTICO I

Introdução

Este documento descreve as características do Trabalho Prático de Banco de Dados II do curso de graduação em Ciência da Computação da Universidade Federal de Itajubá – 1º semestre de 2023.

O trabalho terá duas partes. Na primeira, os alunos devem desenvolver uma aplicação que consome dados de uma API de Dados. Na segunda etapa do trabalho os alunos devem implementar uma aplicação que consome os dados da base criada na primeira etapa por meio de relatórios dinâmicos.

PRIMEIRA PARTE - CONSUMIR DADOS DE UMA API

API de Dados

Segundo Alvarenga et al. (2011), há muitas formas de disponibilizar os dados: eles podem ser publicados em páginas da web, podem ser expostos via uma interface de consulta em um website, ou podem ser acessados diretamente por sistemas eletrônicos via uma API (interface de programação de aplicativo).

Dentre essas formas, o uso de APIs (serviços web), apresenta diversos benefícios, tais como (Kong, 2015):

- Interoperabilidade entre os sistemas, garantindo escalabilidade, facilidade de uso, além de possibilitar atualização de forma simultânea e em tempo real.
- a economia de tempo e custo dos pedidos de acesso à informação
- a possibilidade de cruzar dados de diferentes órgãos gerando novas agregações
- aumento da participação social
- estímulo a inovação
- economia de tempo e dinheiro
- melhora nos serviços governamentais

Na prática, uma API é simplesmente a exposição de uma série de ferramentas, métodos de programação e protocolos, com o objetivo de facilitar a programação de uma aplicação[1].

[1] https://sensedia.com/blog/apis/o-que-sao-apis-parte-1-introducao/

Portanto, uma API de Dados é um serviço web que disponibiliza dados. Para obter os dados é preciso obedecer aos padrões definidos pela interface.

ETL

ETL (Extração, Transformação e Carga) é um processo para extrair dados de um sistema, processá-los, modificá-los e posteriormente inseri-los numa base (banco de dados). A concepção de um processo ETL incide sobre o mapeamento dos atributos dos dados de uma ou várias fontes para os atributos das tabelas do banco de dados.

Metodologia

As etapas do trabalho estão listadas na figura 1. Na **primeira etapa** o grupo deverá estudar o conceito de API e escolher alguma para trabalhar. Cada grupo deve escolher uma API diferente. Para garantir que não haverá grupos utilizando a mesma API, valerá a ordem de apresentação da API pelo grupo no SIGAA. A preferência será por "ordem de chegada". Ou seja, se o seu grupo escolheu uma API que outro grupo já postou, vocês deverão escolher outra. Nessa etapa existe uma restrição importante. É necessário escolher uma API que disponibilize dados via interfaces. Ou seja, não será aceito o uso de APIs que retornam um único arquivo (CSV, JSON) com toda a base de dados.

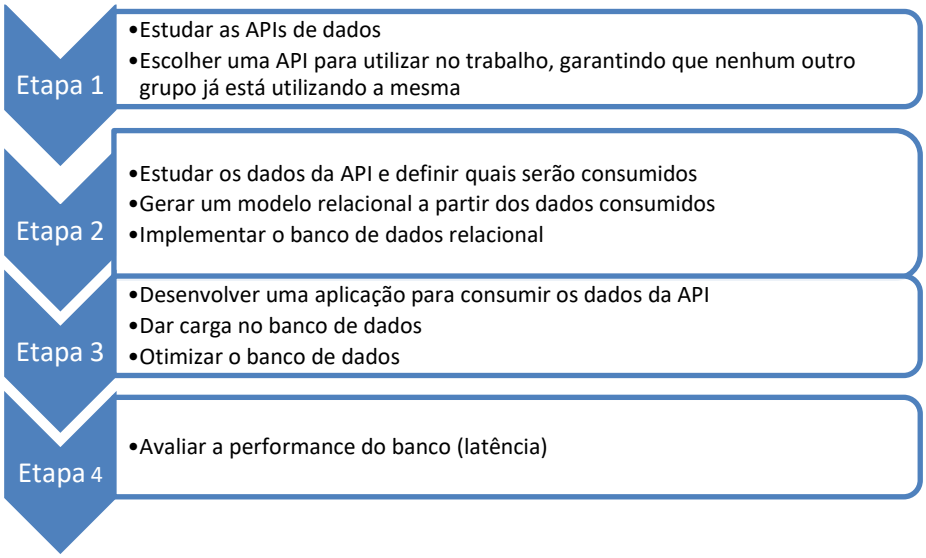

Figura 1 : Etapas do trabalho prático.

APIs BANIDAS – NÃO PODEM SER UTILIZADAS:

- spotify
- cartola
- pokemon

- IBGE (cidades/estados/capitais)
- filmes (qualquer uma com esse tema)

Na **segunda etapa**, o grupo fará o trabalho de engenharia reversa. Em geral, cada API fornece um conjunto enorme de dados. O grupo deverá pensar no conjunto que deseja trabalhar. Os produtos finais dessa etapa serão o modelo relacional e o banco de dados implementado (modelo físico). O SGBD relacional é de livre escolha do grupo. Essa é uma etapa fundamental no projeto. É preciso entender o dado e o que é possível extrair deles. Algumas questões importantes são:

a) Conhecer o processo
b) Definir as informações mais estratégicas do banco
c) Definir os domínios de cada atributo do banco
d) Pensar em como os dados poderão ser utilizados por diferentes níveis de usuários (operacional, estratégico, gestão).

<u>Características desejáveis da API:</u>

- Que seja possível mapear em um conjunto de, ao menos, 5 tabelas do banco relacional
- Que os dados não sejam muito resumidos e que possam ser qualificados. Isso ajudará na hora de gerar o relatório dinâmico.
- Que tenha dados suficientes para estressar o sistema durante os testes de performance.

Na **terceira etapa**, o grupo irá implementar uma aplicação que consome os dados dessa API. A linguagem de programação é livre. Essa aplicação deverá cumprir os seguintes requisitos (figura 2):

1. Acessar a API e fazer o *download* do conjunto de dados a partir de um filtro. Em geral, as APIs retornam dados em algum formato texto (Json, XML, CSV). O grupo pode escolher qual utilizar. A professora prefere JSON ou XML.
2. Extrair os dados do documento texto e dar carga nas tabelas do banco de dados.

É necessário baixar um alto volume de dados. O limite depende do *hardware* utilizado. Baixem dados suficientes para ter uma boa amostra, mas no limite em que o banco retorne os comandos em tempo plausível.

Com o banco populado, o grupo fará as otimizações, como a criação de índices, definição de usuários e de privilégios no banco.

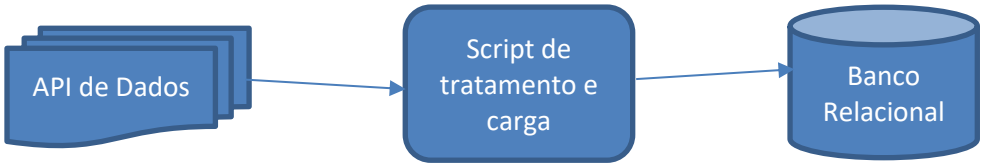

Figura 2 : Atividades da terceira etapa.

Na **quarta etapa**, o grupo escolherá uma consulta custosa no banco para realizar as medidas de performance utilizando o software JMeter[2]. O grupo deverá medir latência nas seguintes condições:

1. Mantém a quantidade de usuários (*threads*) como 1 e aumenta a quantidade de requisições até o teste retornar erro. Esse será o número máximo de requisições suportadas pelo banco para essa consulta.
2. Defina uma quantidade de requisições fixa e aumente a quantidade de usuários até o teste retornar erro. Esse será o número máximo de usuários suportados pelo banco para essa consulta.

Cada análise deverá gerar um gráfico (latência x número de requisições; latência x número de usuários). Vocês devem aumentar a quantidade de *threads* e requisições manualmente. Cada vez que rodarem, façam 30 repetições do teste e extraiam a média. Essa média é o valor que vai para o gráfico. Na Figura 3 há exemplos de gráfico de *latência x número de usuários*.

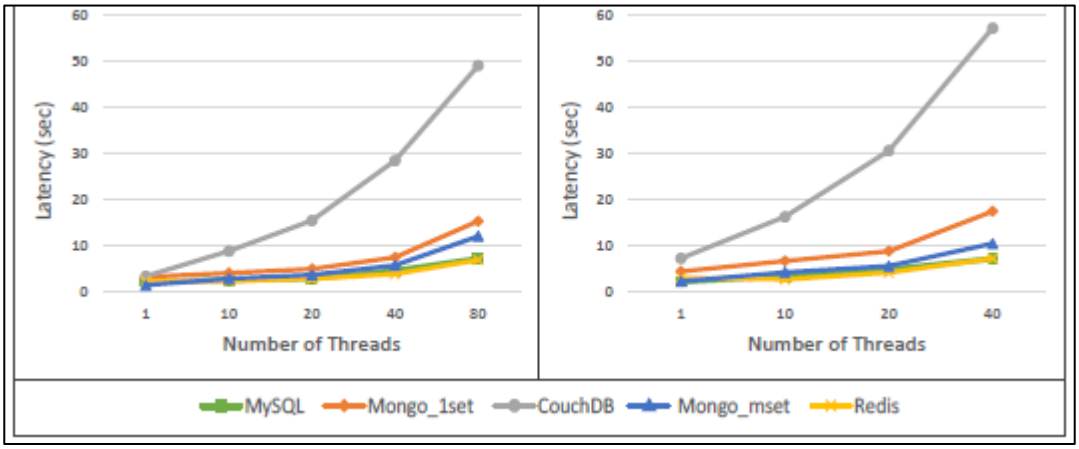

Figura 3 : Exemplo de gráfico que mostra a latência em função do aumento do número de usuários conectados no banco.

[2] https://jmeter.apache.org/

É importante ressaltar que os SGBDs definem, por padrão, um número máximo de requisições e usuários que podem se conectar ao mesmo tempo no banco. Esse número é configurável. Sendo assim, o grupo deve alterar esses valores antes de iniciar os testes.

Para validar os testes, é imprescindível que o grupo informe a arquitetura da máquina utilizada nos testes.

SEGUNDA PARTE – CONSUMIR OS DADOS DO BANCO GERADO

Nesta parte do trabalho, o grupo deverá criar uma aplicação que permita ao usuário acessar um relatório ad-hoc. Relatório ad-hoc é aquele onde o usuário pode **escolher os campos e filtros que deseja**. A consulta é realizada dinamicamente pela aplicação, que entrega somente os dados solicitados pelo usuário. Dessa forma, diferentes níveis de usuários do seu banco possam obter, por meio da interface implementada, um relatório personalizado. A figura 4 apresenta uma interface de criação de um relatório *ad-hoc*.

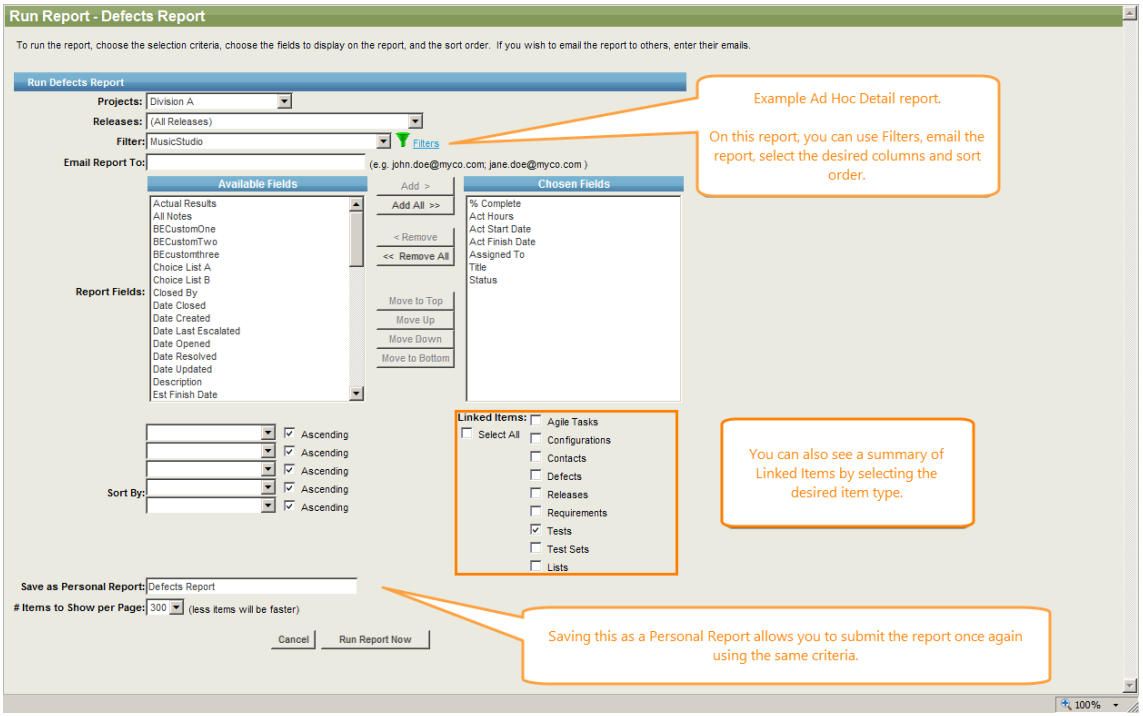

Figura 4 : Exemplo de interface para geração de relatório ad-hoc.
Fonte : https://support.smartbear.com/qacomplete/docs/user/reports/legacy.html.

Nesse trabalho, basta que entreguem os dados em formato tabular. No entanto, quem gerar gráficos e/ou *dashboads* (a partir da consulta *ad-hoc*), terá uma pontuação extra. Mas, atenção! Os gráficos devem ser em função do relatório ad-hoc!

A linguagem de programação é livre, mas é obrigatório o uso de um framework ORM.

ENTREGAS

Primeira Parte: A entrega deve ser feita até o dia **09/06**. Nesta ocasião, o grupo deverá entregar:

a) Relatório do banco de dados (3 pontos)
   - Diagrama Relacional, definição de grupos de usuários e suas permissões, definição de índices e justificativas, definição de views, triggers, procedures, funções e suas justificativas, caso existam.
   - Print com o count das tabelas

b) Resultados dos testes de performance obtidos com o JMeter (3 pontos)
c) Link para o código desenvolvido (github)
d) Um vídeo mostrando a aplicação funcionando. Neste vídeo é importante mostrar a conexão com o banco, funções que fazem a requisição dos dados da API, funções que salvam os dados tratados no banco (4 pontos)

Atenção : Não entregar o código e o vídeo zera o trabalho.

Segunda Parte: Acontecerá nos dias 3 e 5 de Julho, durante as aulas. A ordem de apresentação será sorteada no início da aula do dia 3. No entanto, todos os grupos deverão entregar a apresentação no dia 3 até 18:00h.

Durante a apresentação, o grupo deverá seguir os seguintes tópicos:

- Descrever a API utilizada
- Descrever as ferramentas utilizadas (linguagem, SGBD, frameworks...)
- Contextualização: quem seria o cliente da sua aplicação?
- Modelo Relacional
- Testes de Performance
- Rodar a aplicação em tempo real, demonstrando o relatório ad-hoc funcionando
- Considerações Finais: impressões do grupo sobre o trabalho (dificuldades, estratégias, contribuições...)

Os grupos que rodarem a cliente/servidor ganham ponto extra. Ou seja, aplicação em uma máquina e banco de dados em outra. Não vale o servidor na nuvem. Precisa estar na máquina de algum dos integrantes.

Cada etapa vale 50% da nota referente ao Trabalho Prático 1.

GRUPOS

Grupos de 5 pessoas

DINÂMICA

Será aberto um fórum no SIGAA e os grupos deverão informar:

- Participantes do grupo
- API que irá utilizar
- Durante o semestre há aulas reservadas para consultorias do trabalho. Nessas aulas, o grupo deve apresentar o andamento do trabalho e tirar dúvidas. A presença nas consultorias pontua na entrega final.

Considerações Importantes

- A análise da base de dados é FUNDAMENTAL para que vocês tenham um bom trabalho. É MUITO IMPORTANTE que vocês estudem bem a base antes de escolhê-la.
- NÃO é permitido ao grupo criar dados na base. A aplicação deve girar em torno dos dados disponíveis. Mas, vocês podem combinar dados de diferentes APIs.
- NÃO é permitido baixar dados completos, ou seja, arquivos com o conjunto inteiro de dados.
- NÃO UTILIZAR A API DE DADOS IRÁ ZERAR O TRABALHO
- Não é permitido o uso de softwares como Tableau e Power BI para a implementação da consulta ad-hoc. O relatório deve ser implementado por vocês.
- É praxe dos alunos deixar o teste de performance (JMeter) para o final e isso acarreta perda de nota, uma vez que muitos grupos não conseguem executar o JMeter como deveria. Sugiro que vocês elejam um membro do grupo para aprender usar a ferramenta com maestria.
- Dividam o trabalho entre os integrantes do grupo.
- A professora não interfere na formação dos grupos. Quem precisar, utilizar o próprio fórum.
- Nos dias das apresentações todo grupo deve estar presente.