

Ítems a serem avaliados no Projeto Interdisciplinar

(1 Ponto por item)

Data: 19/06/2017 até as 18:59:59:999

- 1) Documentos de requisitos do sistema;
- 2) MER ou ER;
- 3) Scripts de criação do Banco em MySQL (my_db_name_create.sql);
- 4) Script SQL para popular o banco com massa de dados (INSERT), em todas as tabelas (my_db_name_load.sql);
- 5) Relatórios básicos para extração de tabelas e relacionamentos:
 - a. Scripts para geração de relatórios (com formatação agradável ao usuário) em formato de texto;
 - b. Relatórios de Inteligência de Negócio (no mínimo 5):
 - i. Requisitos (descrição textual);
 - ii. Comando SQL para geração do relatório.
 - c. **OBS: Os relatórios deverão ser implementados utilizando Views.**

OBS: Cada script de criação deverá estar em um arquivo SQL. Assim teríamos por exemplo os arquivos SQL:

- empregados_vw.sql;
- departamentos_vw.sql;
- projeto_vw.sql;
- total_por_projeto_vw.sql;
- func_por_projetos_vw.sql.

- 6) **Usuários e Permissões:** Implementação de Política de Segurança com Usuários e Permissões no mínimo 3 níveis podendo ter mais (Administrador; Analista de Dados; Usuário).
 - a. Descrição da política nome do usuário e respectivas permissões;

- b. Script SQL para criação dos usuários e definição de privilégios.

**OBS: Deve ser entregue em um único arquivo SQL.
my_db_permissions.sql**

7) **Stored Procedures:** Seu sistema não deverá permitir que a camada de aplicação faça chamadas diretas à comandos SQL, logo deverá implementar Procedimentos ou Funções para realizar operações de manipulação de dados (INSERT, DELETE, UPDATE). Consultas diretas são permitidas (Comandos SELECT dentro da aplicação). Ex. Tabela EMPREGADOS

- a. PROC_INSERE_EMP (COD_EMP INT, NOME VARCHAR...) BEGIN ... END;
- b. FUNC_DELETE_EMP (COD_EMP INT, NOME VARCHAR) RETURN INT BEGIN ... END;
 - i. Retornar número de registros excluídos.
- c. PROC_UPDATE_EMP (COD_EMP INT, NOME VARCHAR...)

OBS: Cada script de criação deverá estar em um arquivo SQL. Assim no exemplo teríamos 3 arquivos SQL:

- proc_insere_emp.sql;
- func_delete_emp.sql;
- proc_update_emp.sql;

8) **Triggers:**

- a. Implementar no mínimo 3 regras de negócio utilizando triggers;

9) **Data Pump:** Criar um script que, por meio de uma consulta (SELECT) gera uma saída contendo comandos INSERT para todos os dados de todas as tabelas. Ex.

<pre>SELECT "INSERT INTO EMPREGADOS VALUES (" COD_EMPREGADO "," NOME "," ... </pre>

“),”

FROM EMPREGADOS;

- 10) **Script de Criação:** Criar um script SQL que consolida todos os scripts anteriores, para criar todos os objetos de seu banco e também inserir a massa de dados. A ideia é que o usuário possa carregar objetos e dados executando apenas um arquivo, que faz chamadas aos demais arquivos SQL, em ordem lógica para criação de objetos, e inserção de dados.