

## **Itens a serem avaliados no Projeto Interdisciplinar**

**(1 Ponto por item)**

- 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 VARHCAR...)  
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 VARHCAR...)

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.

```
SELECT "INSERT INTO EMPREGADOS VALUES (" ||  
        COD_EMPREGADO || "," ||  
        NOME || "," || ... ||  
        ");"  
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.