

Você foi contratado para implementar o banco de dados de uma escola de línguas fictícia chamada **Baião de Words Academy**. A escola oferece cursos em diferentes idiomas e organiza as aulas em turmas associadas a professores e níveis de aprendizado (básico, intermediário, avançado). Além disso, é necessário gerenciar as matrículas dos alunos, incluindo notas, cálculo de médias, auditoria e funcionalidades de desmatrícula.

Objetivo

O objetivo desta atividade é implementar um sistema de banco de dados para gerenciar o funcionamento da **Baião de Words Academy**, automatizando as seguintes funcionalidades:

1. Criar as tabelas principais e definir as relações entre elas.
 - a. Criar índice quando necessário
2. Gerenciar os níveis de cada turma e associá-los ao curso e ao professor.
3. Permitir matricular e desmatricular de alunos em turmas.
 - a. Matrícula deve verificar a quantidade de vagas disponíveis na turma
4. Registrar notas e calcular automaticamente a nota final de cada aluno.
5. Implementar auditoria para rastrear alterações no sistema.
6. Adicionar a funcionalidade de aprovação e reprovação dos alunos ao fechar uma turma, com base nas notas.

1. Criação do Banco de Dados e Tabelas

Crie o banco de dados e as tabelas conforme a estrutura abaixo:

- **Tabela Cursos:** Registra os cursos oferecidos pela escola.
cursoid: Identificador único do curso.
nome: Nome do curso.
descricao: Descrição do curso.
duracao: Duração do curso em semestres.
- **Tabela Professores:** Armazena informações dos professores.
professorid: Identificador único do professor.
nome: Nome do professor.
email: E-mail do professor.
cpf: CPF do professor
telefone: Número de telefone do professor.
- **Tabela Alunos:** Armazena informações dos alunos matriculados.
alunoid (INT, PK, AUTO_INCREMENT): Identificador único do aluno.
nome: Nome do aluno.
email: E-mail do aluno.
telefone: Número de telefone do aluno.
endereco: Endereço do aluno.
cpf: cpf do aluno

- **Tabela Turmas:** Define as turmas associadas a cursos, professores e níveis (Básico, Intermediário, Avançado).
turmaid (INT, PK, AUTO_INCREMENT): Identificador único da turma.
nome (VARCHAR(100)): Nome da turma.
cursoid: Referência ao curso oferecido.
professorid: Referência ao professor que ministra a turma.
nível: Nível do curso pode ser 'Básico', 'Intermediário', 'Avançado'.
status: Status da turma (Aberta ou Fechada).
vagas: número de vagas disponíveis na turma (padrão 30)
- **Tabela Matriculas:** Relaciona alunos e turmas, armazenando notas e a nota final.
matriculaid: Identificador único da matrícula.
alunoid: Referência ao aluno matriculado.
turmaid: Referência à turma onde o aluno está matriculado.
datamatricula: Data da matrícula.
nota1: Primeira nota do aluno.
nota2: Segunda nota do aluno.
notafinal: Nota final, calculada pela média das duas notas.
statusmatricula: Status da matrícula, com base na nota final. (Pendente, Aprovado ou Reprovado)
- **Tabela Auditoria:** Registra alterações no sistema para fins de rastreamento.
auditid: Identificador único do registro de auditoria.
tabelaalterada (VARCHAR(50)): Nome da tabela alterada.
acao: Tipo de ação realizada. (INSERIR, ATUALIZAR, EXCLUIR, FECHAR_TURMA)
dataalteracao (DATETIME): Data e hora da alteração.

2. Automatização de Tarefas no BD

1. Aluno
 - a. Não permitir cadastrar / atualizar alunos com cpf iguais
2. Professores
 - a. Não permitir cadastrar / atualizar professores com cpf iguais
3. Curso
 - a. Não permitir cadastrar curso com nomes iguais (pode levar em consideração o case sensitive)
4. Matrícula de Aluno
 - a. Não permitir que o aluno seja matriculado em uma turma que não tenha vaga
 - b. Não permitir que o aluno seja matriculado em um curso / nível que ele já tenha sido aprovado
 - c. Ao matricular o aluno em turma atualizar a quantidade de vagas disponíveis
 - d. Não permitir matricular aluno em turma fechada
 - e. Não permitir matricular o aluno se ele não foi aprovado em nível anterior

- i. Ex.: quer matricular no intermediário mas não fez o básico ainda
 - ii. Ex.: quer matricular no intermediário mas não fez o intermediário ainda
5. Desmatricular Aluno
 - a. Atualizar quantidade de vagas da turma
 - b. Remover registro de matrícula de aluno na turma
 - c. Não permitir desmatricular aluno em turma fechada
6. Calcular Média do aluno
 - a. Ao alterar uma das notas do aluno, calcular a média das notas
 - b. Não permitir alterar notas com valores maiores que 10 e menores que zero
7. Fechar turma
 - a. Ao fechar a turma atualizar o status de cada aluno baseado na média
 - i. ≥ 7 : Aprovado
 - ii. < 7 : Reprovado
 - b. Atualizar o status da turma para fechado
8. Auditoria
 - a. Registrar mudanças baseadas ações no banco de dados

3. Consultas SQL (Pode-se criar uma view)

- Liste todas as turmas de um determinado curso, incluindo os professores e o nível.
- Liste os alunos matriculados em uma turma, com suas notas e a nota final.
- Liste os registros da auditoria para verificar alterações no sistema.
- List todos os alunos e respectivo curso que tem direito a certificado (concluiu os 3 níveis de um curso)
- Liste todos os cursos realizado pelo aluno

4. Decisões

Justifique as suas decisões relacionadas à criação de índices, enums, views, stored procedures, functions e triggers durante o processo de construção do banco de dados. Explique como cada um desses elementos contribui para a eficiência, organização e manutenção do sistema. Além disso, analise como essas escolhas impactam na segurança, na clareza do modelo e na facilidade de futuras modificações ou expansões do banco de dados.

5. Testes

Insira dados fictícios no sistema e realize uma validação detalhada para garantir que todas as funcionalidades implementadas estejam atendendo corretamente aos requisitos especificados. Durante esse processo, execute os seguintes passos:

1. Inserção de Dados Fictícios:

- Crie registros simulados e representativos para cada tabela, levando em consideração os diferentes cenários de uso.
- Certifique-se de que os dados inseridos cubram todas as variações possíveis dos campos (como valores válidos e inválidos, limites de tamanho, etc.).

2. Testes Funcionais:

- Execute as funcionalidades do sistema, como consultas, atualizações, exclusões e operações específicas como a criação de índices, execução de stored procedures, entre outras.
- Verifique se as operações estão sendo realizadas corretamente, sem erros ou falhas, e se o sistema responde conforme esperado.

3. Validação de Requisitos:

- Confirme que os requisitos de negócio e técnicos estabelecidos foram cumpridos. Isso pode incluir verificações como:
 - Se os índices criados estão funcionando conforme o esperado.
 - Se as stored procedures e funções estão sendo executadas corretamente.
 - Se as triggers estão funcionando como gatilhos automáticos de eventos no banco de dados.
 - Se os dados estão sendo manipulados corretamente com base nas variáveis de ambiente e outros parâmetros definidos.