|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| logo_faculdade_impacta_azul-01 | | Banco de Dados  Linguagem SQL  AC4 – Projeto LMS – Modelo Físico | | |
| *Nome do Aluno:**Adriano Maia* | | | | *RA: 1900390* |
| *Nome do Aluno: Bruno de Andrade Barroca* | | | | *RA: 1900794* |
| *Nome do Aluno: Felipe Chin Lau* | | | | *RA: 1900416* |
| *Nome do Aluno: Leticia Marques* | | | | *RA: 1802200* |
| *Turma:* Banco de Dados | *Série:* **2º** | | *Data:* 23/09/2019 | |
| *Professor(a) Responsável:* Gustavo Bianchi Maia | | | | |
| Nota: ( ) Visto do Prof. | | | | |

**1º - Relatório da entrega**

Para conclusão da primeira entrega do projeto LMS, realizamos a seguinte alteração:

Dentro da regra de negócio do modelo original encontramos **“Toda FK terá o nome ID + <tabela de destino>, ou seja: IDCoordenador é FK relacionado à tabela Colaborador, coluna ID”**. Desta forma, durante o processo de criação das tabelas, percebemos que algumas *foreign keys* teriam o mesmo nome.

Exemplo: As *foreign keys* das tabelas Disciplina e DisciplinaOfertada seriam nomeadas de forma idêntica o que traria confusão ao projeto e até mesmo em uma busca futura dentro do banco de dados. Para facilitar esta etapa, decidimos por manter a regra do modelo original + <tabela de origem>, ou seja: a FK IDCoordenador da tabela Disciplina ficaria IDCoordenadorDisciplina, assim melhorando a distribuição das *foreign keys* dentro do projeto.

**Tecnologias:**

Tipo do banco de dados: Microsoft SQL Server

Diagrama visual: Microsoft SQL Server

**2º - Diagrama visual das tabelas (SSMS, Myadmin, Erwin, Visio.... )**

* + - **Formato da entrega 🡪 imagem ( .JPG, etc )**
    - **Os nomes das tabelas, colunas e relacionamentos devem estar legíveis.**
    - **São permitidas múltiplas imagens (caso o modelo não esteja visível em apenas uma imagem ).**

(outro arquivo em PDF).

**3º - Código de criação ( create / alter )**

/\*

Faculdade Impacta de Tecnologia

Projeto LMS (Learning Management System)

Curso: Banco de Dados

Disciplina: Linguagem SQL

\*/

------------------------------------------------------------------------------------------------------

CREATE DATABASE ProjetoLMS;

GO

USE ProjetoLMS;

------------------------------------------------------------------------------------------------------

CREATE TABLE Usuario(

ID INT NOT NULL IDENTITY (1,1),

[Login] VARCHAR (50) NOT NULL,

Senha VARCHAR (50) NOT NULL,

DtExpedição DATE NOT NULL CONSTRAINT DF\_DataExpedição DEFAULT '01/01/1900',

CONSTRAINT Pk\_Usuario PRIMARY KEY (ID),

CONSTRAINT UQ\_LoginUsuario UNIQUE([Login]),

);

GO

------------------------------------------------------------------------------------------------------

CREATE TABLE Coordenador(

ID INT NOT NULL IDENTITY (1,1),

Nome VARCHAR (50) NOT NULL,

ID\_Usuario INT NOT NULL,

Email VARCHAR (50) NOT NULL,

Celular TINYINT,

CONSTRAINT Pk\_Coordenador PRIMARY KEY (ID),

CONSTRAINT UQ\_EmailCoordenador UNIQUE(Email),

CONSTRAINT UQ\_CelularCoordenador UNIQUE(Celular),

CONSTRAINT Ck\_CelularCoordenador CHECK ( Celular LIKE '[0-9][0-9][0-9]) [0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),

CONSTRAINT fk\_CoordenadorUsuario FOREIGN KEY (ID\_USUARIO) REFERENCES Usuario (ID)

);

GO

------------------------------------------------------------------------------------------------------

CREATE TABLE Aluno(

ID INT NOT NULL IDENTITY (1,1),

Nome VARCHAR (50) NOT NULL,

ID\_Usuario INT NOT NULL,

Email VARCHAR (50) NOT NULL,

Celular TINYINT,

RA TINYINT NOT NULL,

Foto VARCHAR (50) NULL,

CONSTRAINT Pk\_Aluno PRIMARY KEY (ID),

CONSTRAINT UQ\_EmailAluno UNIQUE(Email),

CONSTRAINT UQ\_CelularAluno UNIQUE(Celular),

CONSTRAINT Ck\_CelularAluno CHECK ( Celular LIKE '[0-9][0-9][0-9]) [0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),

CONSTRAINT fk\_AlunoUsuario FOREIGN KEY (ID\_USUARIO) REFERENCES Usuario (ID)

);

GO

------------------------------------------------------------------------------------------------------

CREATE TABLE Professor(

ID INT NOT NULL IDENTITY(1,1),

Id\_usuario INT NOT NULL,

Email VARCHAR(100),

Celular TINYINT,

Apelido VARCHAR(50)

CONSTRAINT PK\_Professor PRIMARY KEY (ID),

CONSTRAINT FK\_IDId\_usuarioProfessor FOREIGN KEY (Id\_usuario) REFERENCES Usuario(ID),

CONSTRAINT UQ\_EmailProfessor UNIQUE(Email),

CONSTRAINT UQ\_CelularProfessor UNIQUE(Celular)

);

GO

------------------------------------------------------------------------------------------------------

CREATE TABLE Disciplina(

ID INT NOT NULL IDENTITY(1,1),

Nome VARCHAR(100) NOT NULL,

[Data] DATE NOT NULL CONSTRAINT DF\_DataDisciplina DEFAULT(GETDATE()),

[Status] VARCHAR(7) NOT NULL CONSTRAINT DF\_StatusDisciplina DEFAULT 'Aberta',

PlanoDeEnsino VARCHAR(200) NOT NULL,

CargaHoraria INT NOT NULL,

Competencias VARCHAR(200) NOT NULL,

Habilidades VARCHAR(200) NOT NULL,

Ementa VARCHAR(5000) NOT NULL,

ConteudoProgramatico VARCHAR(5000) NOT NULL,

BibliograficaBasica VARCHAR(3000) NOT NULL,

BibliograficaComplementar VARCHAR(3000) NOT NULL,

PercentualPratico TINYINT NOT NULL,

PercentualTeorico TINYINT NOT NULL,

IdCoordenador INT NOT NULL

CONSTRAINT PK\_Disciplina PRIMARY KEY(ID),

CONSTRAINT FK\_IDCoordenadorDisciplina FOREIGN KEY (IdCoordenador) REFERENCES Coordenador(ID),

CONSTRAINT UQ\_NomeDisciplina UNIQUE(Nome),

CONSTRAINT CK\_StatusDisciplina CHECK([Status] IN('Aberta','Fechada')),

CONSTRAINT CK\_PercentualPraticoDisciplina CHECK(PercentualPratico >= 0 AND PercentualPratico <= 100),

CONSTRAINT CK\_PercentualTeoricoDisciplina CHECK(PercentualTeorico >= 0 AND PercentualTeorico <= 100)

);

GO

------------------------------------------------------------------------------------------------------

CREATE TABLE Curso(

ID INT NOT NULL IDENTITY(1,1),

NomeCurso VARCHAR(30) NOT NULL,

CONSTRAINT PK\_IdCurso PRIMARY KEY (ID),

CONSTRAINT UQ\_NomeCurso UNIQUE (NomeCurso)

);

GO

------------------------------------------------------------------------------------------------------

CREATE TABLE DisciplinaOfertada(

ID INT NOT NULL IDENTITY(1,1),

IdCoordenador INT NOT NULL,

DtInicioMatricula DATE NULL,

DtFimMatricula DATE NULL,

IdDisciplina INT NOT NULL,

IdCurso INT NOT NULL,

Ano SMALLINT NOT NULL,

Semestre TINYINT NOT NULL,

Turma CHAR(1) NOT NULL,

IdProfessor INT NULL,

Metodologia VARCHAR(3000) NULL,

Recursos VARCHAR(3000) NULL,

CriterioAvaliacao VARCHAR(3000) NULL,

PlanoDeAulas VARCHAR(3000) NULL

CONSTRAINT PK\_DisciplinaOfertada PRIMARY KEY (ID),

CONSTRAINT FK\_IDCoordenadorDisciplinaOfertada FOREIGN KEY (IdCoordenador) REFERENCES Coordenador(ID),

CONSTRAINT FK\_IDDisciplinaDisciplinaOfertada FOREIGN KEY (IdDisciplina) REFERENCES Disciplina(ID),

CONSTRAINT FK\_IDCursoDisciplinaOfertada FOREIGN KEY (IdCurso) REFERENCES Curso(ID),

CONSTRAINT PK\_IDProfessorDisciplinaOfertada FOREIGN KEY (IdProfessor) REFERENCES Professor(ID),

CONSTRAINT CK\_AnoDisciplinaOfertada CHECK(Ano BETWEEN 1900 AND 2100),

CONSTRAINT CK\_SemestreDisciplinaOfertada CHECK(Semestre BETWEEN 1 AND 2),

CONSTRAINT CK\_TurmaDisciplinaOfertada CHECK(Turma LIKE '[A-Z]'),

CONSTRAINT UQ\_ValidaOfertaDisciplinaOfertada UNIQUE(idDisciplina, idCurso, Ano, Semestre, Turma)

);

GO

------------------------------------------------------------------------------------------------------

CREATE TABLE SolicitacaoMatricula(

ID INT NOT NULL IDENTITY(1,1),

IdAluno INT NOT NULL,

IdDisciplinaOferdatada INT NOT NULL,

DtSolicitacao DATE NOT NULL CONSTRAINT DF\_DtSolicitacao DEFAULT (GETDATE()),

--"IdCoordenador não é obrigatório de ser preenchido, porém, quando o coordenador aprová-la (ie: alterar o status da mesma), seu ID deve ser preenchido.-->

IdCoordenador INT NULL,

Status VARCHAR(10) NOT NULL CONSTRAINT DF\_Status DEFAULT ('Solicitada'),

CONSTRAINT CK\_Status CHECK (Status IN ('Solicitada', 'Aprovada', 'Rejeitada', 'Cancelada')),

-- Alterado FK\_IdAluno para FK\_IdAlunoSolicitacaoMatribula --

CONSTRAINT FK\_IdAlunoSolicitacaoMatricula FOREIGN KEY (IdAluno) REFERENCES Aluno (ID),

CONSTRAINT FK\_IdDisciplinaOfertada FOREIGN KEY (IdDisciplinaOferdatada) REFERENCES DisciplinaOfertada (ID),

);

GO

------------------------------------------------------------------------------------------------------

CREATE TABLE Atividade(

ID INT NOT NULL IDENTITY(1,1),

Titulo VARCHAR(50) NOT NULL,

Descricao VARCHAR(3000) NOT NULL,

Conteudo VARCHAR(3000) NOT NULL,

Tipo VARCHAR(14) NOT NULL CONSTRAINT CK\_Tipo CHECK (Tipo IN ('Resposta Aberta','Teste')),

Extras VARCHAR(30) NULL,

IdProfessor INT NOT NULL

CONSTRAINT Pk\_Atividade PRIMARY KEY (ID), -- NECESSARIO INCLUIR PK

CONSTRAINT FK\_IdProfessor FOREIGN KEY (IdProfessor) REFERENCES Professor (ID)

);

GO

------------------------------------------------------------------------------------------------------

------------------------------------------------------------------------------------------------------

CREATE TABLE AtividadeVinculada(

ID INT NOT NULL IDENTITY(1,1),

IdAtividade INT NOT NULL,

IdProfessor INT NOT NULL,

IdDisciplinaOferdatada INT NOT NULL,

Rotulo VARCHAR(30) NOT NULL,

Status VARCHAR(30) NOT NULL,

DtInicioRespostas DATE NOT NULL,

DtFimRespostas DATE NOT NULL,

CONSTRAINT Pk\_AtividadeVinculada PRIMARY KEY (ID),

CONSTRAINT FKIdAtividade FOREIGN KEY (IdAtividade) REFERENCES Atividade (ID),

CONSTRAINT FKIdProfessor FOREIGN KEY (IdProfessor) REFERENCES Professor (ID),

CONSTRAINT FKIdDisciplinaOfertada FOREIGN KEY (IdDisciplinaOferdatada) REFERENCES DisciplinaOfertada (ID),

CONSTRAINT UQ\_Rotulo UNIQUE (Rotulo),

CONSTRAINT CK\_StatusAtvVcl

CHECK ([Status] in ('Disponibilizada','Aberta', 'Fechada', 'Encerrada', 'Prorrogada'))

);

GO

------------------------------------------------------------------------------------------------------

CREATE TABLE Entrega(

ID INT NOT NULL IDENTITY(1,1),

IdAluno INT NOT NULL ,

IdAtividadevinculada INT NOT NULL,

Titulo VARCHAR(500),

Resposta VARCHAR(500),

DtEntrega DATE NOT NULL CONSTRAINT DtEntrega DEFAULT (GETDATE()),

[Status] VARCHAR(10) NOT NULL CONSTRAINT DF\_StatusEtg DEFAULT 'Entregue',

IdProfessor INT NOT NULL,

Nota numeric(4, 2),

DtAvaliacao DATE,

Obs VARCHAR(500),

CONSTRAINT Pk\_Entrega PRIMARY KEY (ID),

CONSTRAINT FK\_Id\_Aluno FOREIGN KEY (IdAluno) REFERENCES Aluno (ID),

CONSTRAINT FK\_Id\_Atividadevinculada FOREIGN KEY (IdAtividadevinculada) REFERENCES Atividadevinculada (ID),

CONSTRAINT FK\_Id\_Professor FOREIGN KEY (IdProfessor) REFERENCES Professor (ID),

CONSTRAINT CK\_StatusEtg

CHECK ([Status] in ('Entregue','Corrigido'))

);

GO

------------------------------------------------------------------------------------------------------

CREATE TABLE Mensagem(

ID INT NOT NULL IDENTITY(1,1),

IdAluno INT NOT NULL,

IdProfessor INT NOT NULL,

Assunto VARCHAR(500) NOT NULL,

Referência VARCHAR(500) NOT NULL,

Conteúdo VARCHAR(500) NOT NULL,

Status VARCHAR(10) NOT NULL,

DtEnvio DATE NOT NULL CONSTRAINT DtEnvio DEFAULT (GETDATE()),

DtResposta DATE,

Resposta DATE,

CONSTRAINT Pk\_Mensagem PRIMARY KEY (ID),

CONSTRAINT FKId\_Aluno FOREIGN KEY (IdAluno) REFERENCES Aluno (ID),

CONSTRAINT FKId\_Professor FOREIGN KEY (IdProfessor) REFERENCES Professor (ID),

CONSTRAINT CK\_StatusEMsg

CHECK ([Status] in ('Entregue', 'Lido', 'Respondido'))

);

/\*

Projeto LMS

Alterações realizadas:

- Alteração do nome de ProjetoLMSVTESTE para ProjetoLMS

- Criar a PRIMARY KEY da tabela Atividade para dar continuidade na cração da tabela AtividadeVinculada

- Duvidas na criação da tabela Mensagem: "Quando o destinatário ler a mensagem ele deve alterar o status para Lido; e quando ela for respondida, preencher a DtRespostae o campo resposta; além de alterar o status para Respondido."

\*/

------------------------------------------------------------------------------------------------------