| Desenho de personagem de desenhos animados com texto preto sobre fundo branco  Descrição gerada automaticamente com confiança média | Ano Letivo:  20251 | Turno:  Noturno | Local:  A definir |
| --- | --- | --- | --- |
| Curso:  ENGENHARIA DA COMPUTAÇÃO | | Turma:  3ª Fase |
| Disciplina:  BANCO DE DADOS II | | Professor(a):  JORGE LUIZ DA SILVA | |

**TRABALHO FINAL COM BASE EM METODOLOGIAS ATIVAS DE APRENDIZAGEM**

**Projeto de banco de dados para um sistema de << Insira o nome do sistema aqui >>**

Nome do aluno 1 + conta github

Nome do aluno 2 + conta github

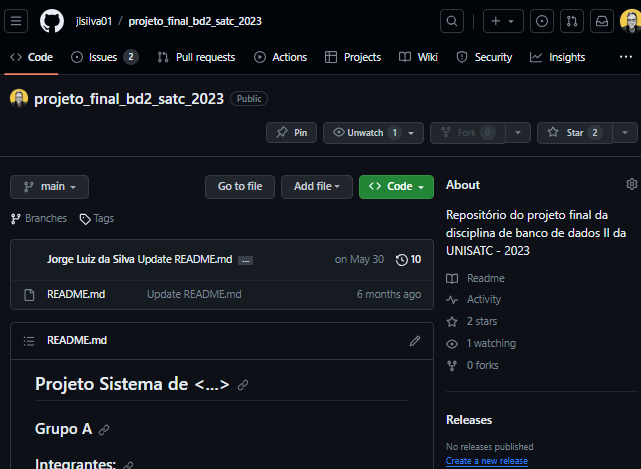
Nome do aluno 3 + conta github

...

Criciúma, 26/05/2025

**URL do projeto no GitHub**

<https://github.com/jlsilva01/projeto-final-bd2-satc-sistema-xyz>

****

**Modelo ER Físico**

**Diagrama

Descrição gerada automaticamente**

**Dicionário de Dados**

| **Tabela** | Avaliação | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Descrição** | Tabela responsável por armazenar os dados das avaliações criadas pelos professores | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| cd\_avaliacao | | int | 1 – 2.147.483.647 | NOT NULL | X |  | Código de identificador da avaliação |
| nm\_avaliacao | | varchar(200) | 10 - 200 | NOT NULL |  |  | Nome da avaliação |
|  | |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
| Idx\_dt\_avaliacao | | X | X | X | cd\_avaliacao, dt\_avaliacao | | |
| Idx\_tipo | |  | X |  | cd\_avaliacao, dt\_avaliacao | | |

| **Tabela** | Avaliação | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Descrição** | Tabela responsável por armazenar os dados das avaliações criadas pelos professores | | | | | | |
| **Atributos** | | | | | | | |
| **Nome da Coluna** | | **Tipo do Dado** | **Valor min e max** | **Nulidade** | **PK** | **FK** | **Descrição** |
| cd\_avaliacao | | int | 1 – 2.147.483.647 | NOT NULL | X |  | Código de identificador da avaliação |
| nm\_avaliacao | | varchar(200) | 10 - 200 | NOT NULL |  |  | Nome da avaliação |
|  | |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |
|  | |  |  |  |  |  |  |
| **Índice** | | | | | | | |
| **Nome do índice** | | **Clustered** | **NonClustered** | **Unique** | **Colunas** | | |
| Idx\_dt\_avaliacao | | X | X | X | cd\_avaliacao, dt\_avaliacao | | |
| Idx\_tipo | |  | X |  | cd\_avaliacao, dt\_avaliacao | | |

**Script dos comandos DDL para criação do Banco de dados (mínimo 8 tabelas)**

/\*==============================================================\*/

/\* Table: aluno \*/

/\*==============================================================\*/

create table aluno (

cd\_aluno int identity,

nm\_aluno varchar(100) null,

email varchar(100) null,

constraint pk\_aluno primary key (cd\_aluno)

)

go

/\*==============================================================\*/

/\* Table: avaliacao \*/

/\*==============================================================\*/

create table avaliacao (

cd\_avaliacao int identity,

ds\_avaliacao varchar(100) null,

dt\_abertura datetime null,

dt\_fechamento datetime null,

constraint pk\_avaliacao primary key (cd\_avaliacao)

)

go

/\*==============================================================\*/

/\* Table: avaliacao\_aluno \*/

/\*==============================================================\*/

create table avaliacao\_aluno (

cd\_avaliacao\_aluno int identity,

cd\_aluno int null,

cd\_avaliacao int null,

ds\_avaliacao\_aluno varchar(100) null,

dt\_inicio datetime null,

dt\_fim datetime null,

constraint pk\_avaliacao\_aluno primary key (cd\_avaliacao\_aluno)

)

go

**Script que popula as tabelas do Banco de dados (mínimo 10 itens principais)**

/\* ADICIONA ALUNOS \*/

--INSERT INTO aluno (nm\_aluno, email)

-- VALUES ('FULANO DE TAL', 'fulano@satc.edu.br');

--GO

insert into aluno (nm\_aluno, email)

select 'MARISA MELO OLIVEIRA', 'email@satc.edu.br' ;

insert into aluno (nm\_aluno, email)

select 'MURILO CARVALHO CARDOSO', 'email@satc.edu.br' ;

insert into aluno (nm\_aluno, email)

select 'VINICIUS ROCHA RODRIGUES', 'email@satc.edu.br' ;

/\* ADICIONA AS ALTERNATIVAS DE CADA QUESTAO\*/

INSERT INTO questao\_item (cd\_questao, ds\_questao\_item, is\_correta) VALUES (1, 'Dados / Comunicação / Conhecimento' ,0);

INSERT INTO questao\_item (cd\_questao, ds\_questao\_item, is\_correta) VALUES (1, 'Análise / Dados / Conhecimento' ,0);

INSERT INTO questao\_item (cd\_questao, ds\_questao\_item, is\_correta) VALUES (1, 'Informação / Compreensão / Comunicação' ,0);

INSERT INTO questao\_item (cd\_questao, ds\_questao\_item, is\_correta) VALUES (1, 'Dados / Informação / Conhecimento' ,1);

GO

INSERT INTO questao\_item (cd\_questao, ds\_questao\_item, is\_correta) VALUES (2, 'F, V, F, F, V', 1);

INSERT INTO questao\_item (cd\_questao, ds\_questao\_item, is\_correta) VALUES (2, 'V, F, F, V, F', 0 );

INSERT INTO questao\_item (cd\_questao, ds\_questao\_item, is\_correta) VALUES (2, 'V, V, V, F, V', 0);

INSERT INTO questao\_item (cd\_questao, ds\_questao\_item, is\_correta) VALUES (2, 'F, F, V, V, F', 0);

GO

**Principais consultas mapeadas baseadas em regras de negócio (mínimo 6)**

--13. (Aberta) Utilizando a linguagem SQL, execute consultas que possam responder a seguinte questão: Qual(is) Carro(s) possui(em) mais sinistros cadastrados?

select s.placa, COUNT(\*)

from sinistro s inner join carro c

on s.placa = c.placa

group by s.placa

order by COUNT(\*) desc

--14. (Aberta) Quantos clientes cadastrados possuem letra “a” no meio e não no final?

select \* --count(\*)

from cliente

where nome like '%a%'

and nome not like '%a'

--15. (Aberta) Quantos clientes não possuem apólice?

select nome, cod\_apolice --\* --count(\*)

from cliente c left join apolice a

on c.cod\_cliente = a.cod\_cliente

where cod\_apolice is null

select nome, cod\_apolice

from cliente c inner join apolice a on c.cod\_cliente = a.cod\_cliente

--16. (Aberta) Quais carros não possuem apólice?

select \* --count(\*)

from carro c left join apolice a on c.placa = a.placa

where cod\_apolice is null

select \*

from apolice

where placa not in (select placa from carro)