|  |  |
| --- | --- |
|  | **INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA**  **DE SÃO PAULO - Campus Bragança Paulista**  **Curso: Técnico em Informática Integrado ao Ensino Médio**  **Disciplina: Banco de Dados 2 Prof. Wilson Vendramel Integrantes do Grupo:**  **Nome: Gabriel ferreira da silva BP: 3050475**  **Nome: RA:**  **Nome: RA: Nome: RA:** |

**Trabalho 3 de Banco de Dados**

Instruções Gerais. Leia com bastante atenção.

1. Este trabalho contém exercícios conceituais e práticos, e pode ser realizado em grupo de até quatro estudantes;

1. O grupo 1 de exercícios constitui 70% da Nota do 3º Bimestre e tem o propósito de projetar e implementar um banco de dados relacional do Projeto Integrador. O grupo 2 de exercícios, por sua vez, constitui 30% da Nota do 3º Bimestre e seu objetivo é apresentar o banco de dados modelado e implementado;

1. Este documento deve ser entregue em um único arquivo PDF em formato organizado e legível. Além disto, os exercícios devem ser apresentados conforme enunciados e seções; cada artefato não apresentado na ordem prevista e no seu devido espaço vai acarretar a anulação do exercício correspondente. Se for favorecer a visualização do artefato, cabe ressaltar que links de acesso a arquivos de imagem podem ser inseridos em tal documento. Deve ficar claro que os exercícios com artefatos ilegíveis serão anulados;

1. O trabalho em foco deve ser entregue como tarefa na plataforma Moodle até às 23h59 de 18/09//2025. A entrega por um colega do grupo já é suficiente, assim sendo, deve ficar claro que não há a necessidade de todos os integrantes da equipe entregarem tal tarefa;

1. O trabalho deve ser verificado e validado antes de ser entregue, dado que não serão aceitas novas entregas, parcial ou totalmente após o prazo final;

1. O trabalho deve ser apresentado em aproximadamente 10 minutos na aula do dia 18/09, conforme ordem previamente estabelecida na aula do dia 11/09;

1. Os materiais disponibilizados auxiliam a realização das atividades do projeto;

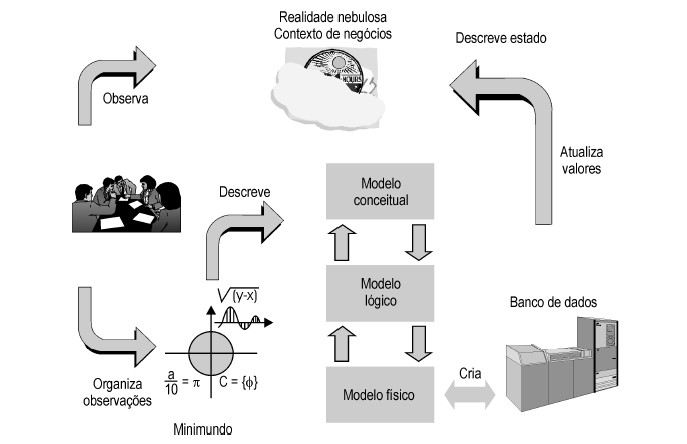
1. Os trabalhos suspeitos de plágio serão devidamente anulados e zerados. Cada exercício que apresentar os mesmos artefatos (entregáveis) em dois ou mais projetos será anulado e zerado de maneira apropriada;

1. Exceções sobre as instruções supracitadas devem ser tratadas com o próprio professor antecipadamente.

------------------------------------------------------------------------------------------------------------------------------------------------------------ **Grupo 1: Projeto e Implementação de Banco de Dados Relacional do Projeto Integrador (7,0 pontos)**

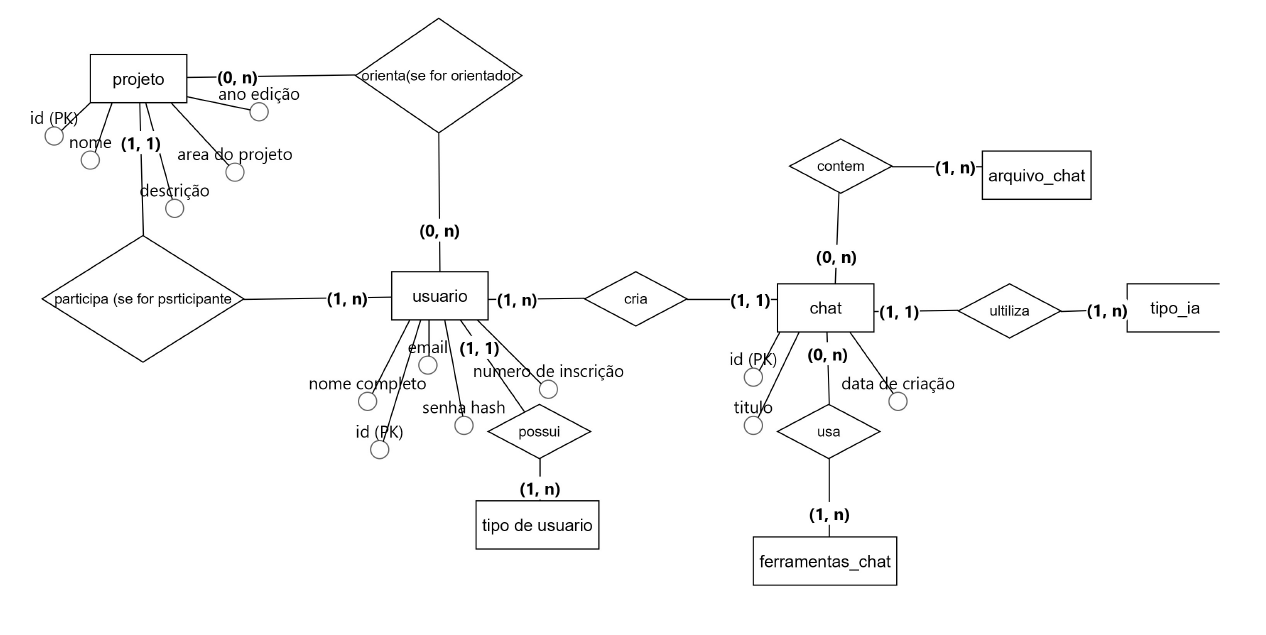
Os exercícios deste grupo se baseiam nas atividades de Projeto de Banco de Dados apresentado por Machado (2014), conforme mostra a Figura 1. Cabe destacar que deve haver uma relação lógica entre os exercícios e que as inconsistências de artefatos anteriores influenciam os posteriores.

Figura 1: Projeto de Banco de Dados



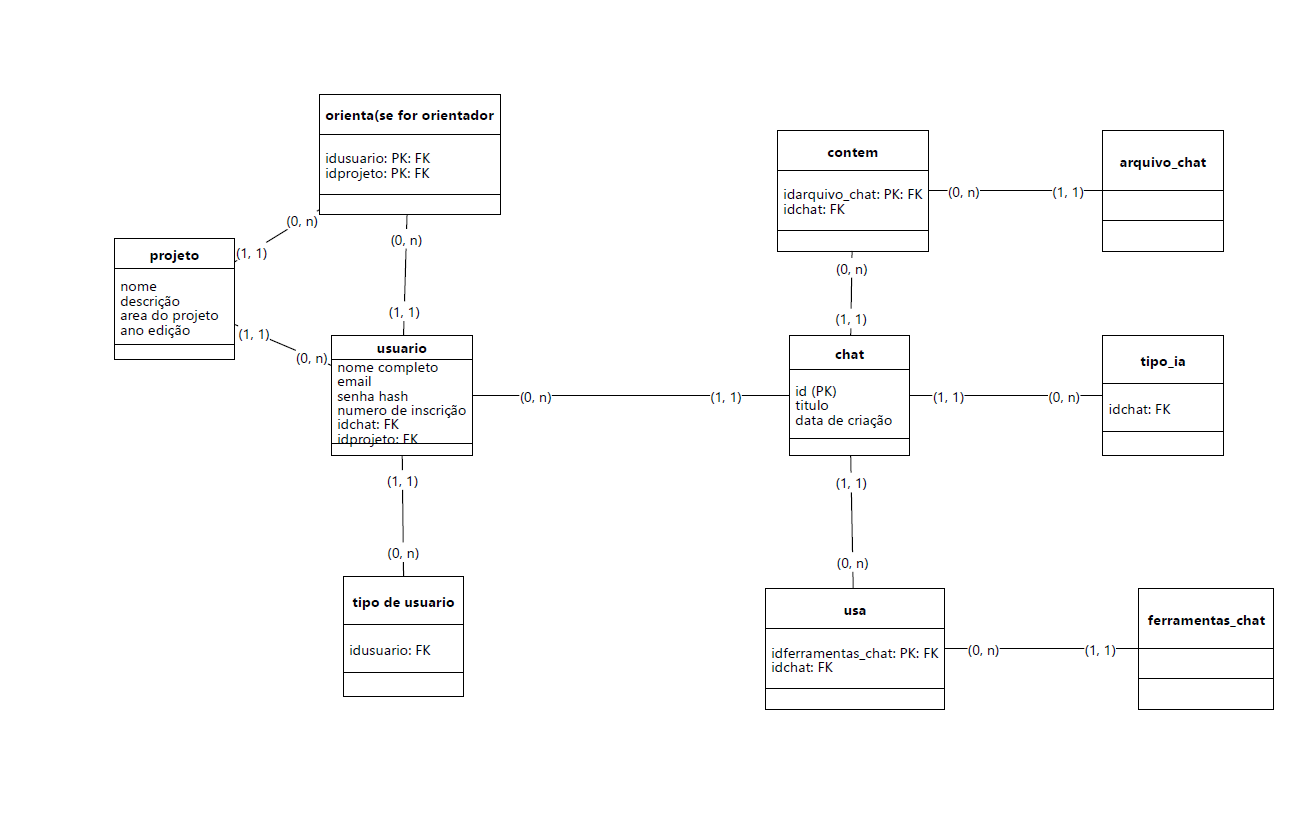
Fonte: Machado (2014, p. 17).

1. Levando em consideração a especificação do software do projeto integrador, organize as ideias a partir do processo de abstração e represente o Modelo Conceitual dos Dados via Diagrama Entidade-Relacionamento (DER) na ferramenta BR Modelo. Atenção com a representação do atributo identificador nas entidades e a cardinalidade dos relacionamentos, especialmente, a cardinalidade N:N. (1,0 ponto) [Modelo Conceitual de Dados ou link de acesso à imagem aqui]



1. Com base no modelo conceitual de dados, represente o Modelo Lógico de Dados por meio de estruturas de tabelas (abordagem relacional) na ferramenta BR Modelo cujo diagrama deve converter as entidades em tabelas. Cabe ressaltar que tal diagrama deve representar as chaves primárias e estrangeiras, assim como as cardinalidades dos relacionamentos, especialmente, as tabelas geradas a partir da cardinalidade N:N. (0,5 ponto)

[Modelo Lógico de Dados ou link de acesso à imagem aqui]



1. Com base no modelo lógico de dados e utilizando o MySQL Workbench, crie um banco de dados e

implemente as tabelas via comandos SQL. Para cada tabela criada, se deve copiar o script no determinado quadro e, na sequência, exibir a estrutura da tabela (print do resultado do comando *describe*). Atenção ao definir apropriadamente as propriedades das colunas (primary key, foreign key, unique, default, not null, auto\_increment). Vale ressaltar que os scripts e prints de cada tabela devem estar legíveis e organizados sequencialmente. (1,5 ponto)

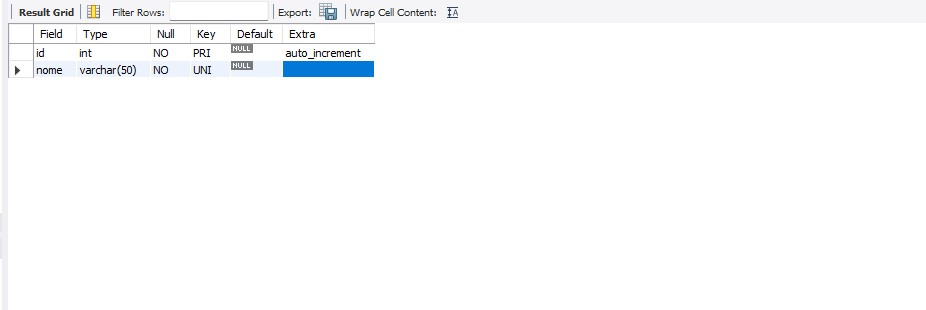
Script de criação da tabela tipos\_usuario (copiar o script; não tirar print).

create table if not exists tipos\_usuario ( -- se é orientador, estudante e etc... id int auto\_increment primary key,

nome varchar(50) not null unique

);

Print da estrutura da tabela tipos\_usuarios.

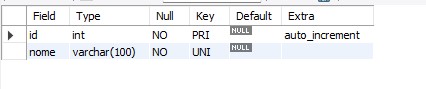


Script de criação da tabela tipos\_ia (copiar o script; não tirar print).

create table if not exists tipos\_ia ( id int auto\_increment primary key, -- personalização basica de personalidade da ia pra agradar melhor cada tipo nome varchar(100) not null unique

);

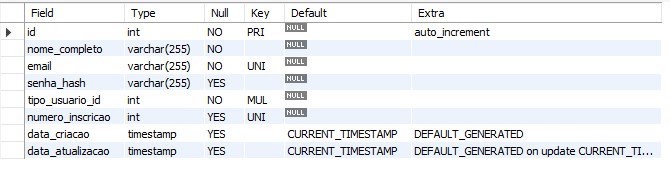
Print da estrutura da tabela tipos\_ia.



Script de criação da tabela usuarios (copiar o script; não tirar print).

|  |
| --- |
| create table if not exists usuarios ( id int auto\_increment primary key, nome\_completo varchar(255) not null, email varchar(255) unique not null, senha\_hash varchar(255), -- vazio se for visitante tipo\_usuario\_id int not null,  numero\_inscricao int unique, -- se for participante da bragantec data\_criacao timestamp default current\_timestamp, -- pega a data atual  data\_atualizacao timestamp default current\_timestamp on update current\_timestamp, foreign key (tipo\_usuario\_id) references tipos\_usuario(id)  ); |

Print da estrutura da tabela usuarios.



Script de criação da tabela projetos (copiar o script; não tirar print).

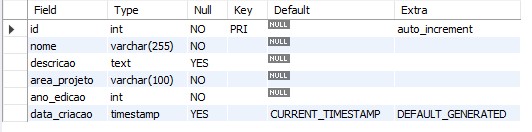
create table if not exists projetos ( -- projeto se for participante da bragantec id int auto\_increment primary key, nome varchar(255) not null,

descricao text, -- text pois pode ser uma descrição muito grande area\_projeto varchar(100) not null,

ano\_edicao int not null, -- se for participar mais de um ano, vai saber data\_criacao timestamp default current\_timestamp

);

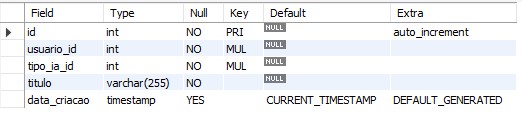
Print da estrutura da tabela projetos.



Script de criação da tabela chats (copiar o script; não tirar print).

|  |
| --- |
| create table if not exists chats ( -- historico de chats, ultil se o usuario quiser voltar algo id int auto\_increment primary key,  usuario\_id int not null, tipo\_ia\_id int not null, titulo varchar(255) not null, data\_criacao timestamp default current\_timestamp, foreign key (usuario\_id) references usuarios(id) on delete cascade, foreign key (tipo\_ia\_id) references tipos\_ia(id)  ); |

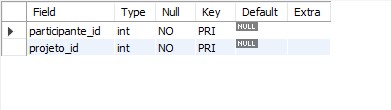
Print da estrutura da tabela chats.



Script de criação da tabela participantes\_projetos (copiar o script; não tirar print).

|  |
| --- |
| create table if not exists participantes\_projetos ( -- tabela de junção pra assosiar projetos e participantes participante\_id int not null, projeto\_id int not null,  primary key (participante\_id, projeto\_id), foreign key (participante\_id) references usuarios(id) on delete cascade, foreign key (projeto\_id) references projetos(id) on delete cascade  ); |

Print da estrutura da tabela participantes\_projetos.



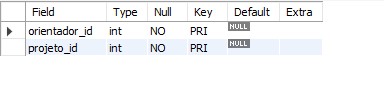
Script de criação da tabela orientadores\_projetos (copiar o script; não tirar print).

create table if not exists orientadores\_projetos ( -- mesma coisa, assosica os projetos a seus orientadores orientador\_id int not null, projeto\_id int not null,

primary key (orientador\_id, projeto\_id), foreign key (orientador\_id) references usuarios(id) on delete cascade, foreign key (projeto\_id) references projetos(id) on delete cascade

);

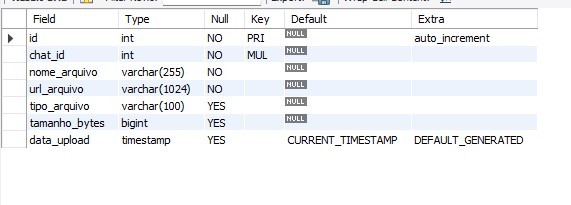
Print da estrutura da tabela orientadores\_projetos.



Script de criação da tabela arquivos\_chat (copiar o script; não tirar print).

|  |
| --- |
| create table if not exists arquivos\_chat ( -- é nessesario para o historico de chats id int auto\_increment primary key, chat\_id int not null,  nome\_arquivo varchar(255) not null, url\_arquivo varchar(1024) not null, tipo\_arquivo varchar(100), tamanho\_bytes bigint,  data\_upload timestamp default current\_timestamp,  foreign key (chat\_id) references chats(id) on delete cascade  ); |

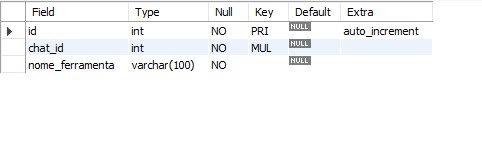
Print da estrutura da tabela arquivos\_chat.



Script de criação da tabela ferramentas\_chat (copiar o script; não tirar print).

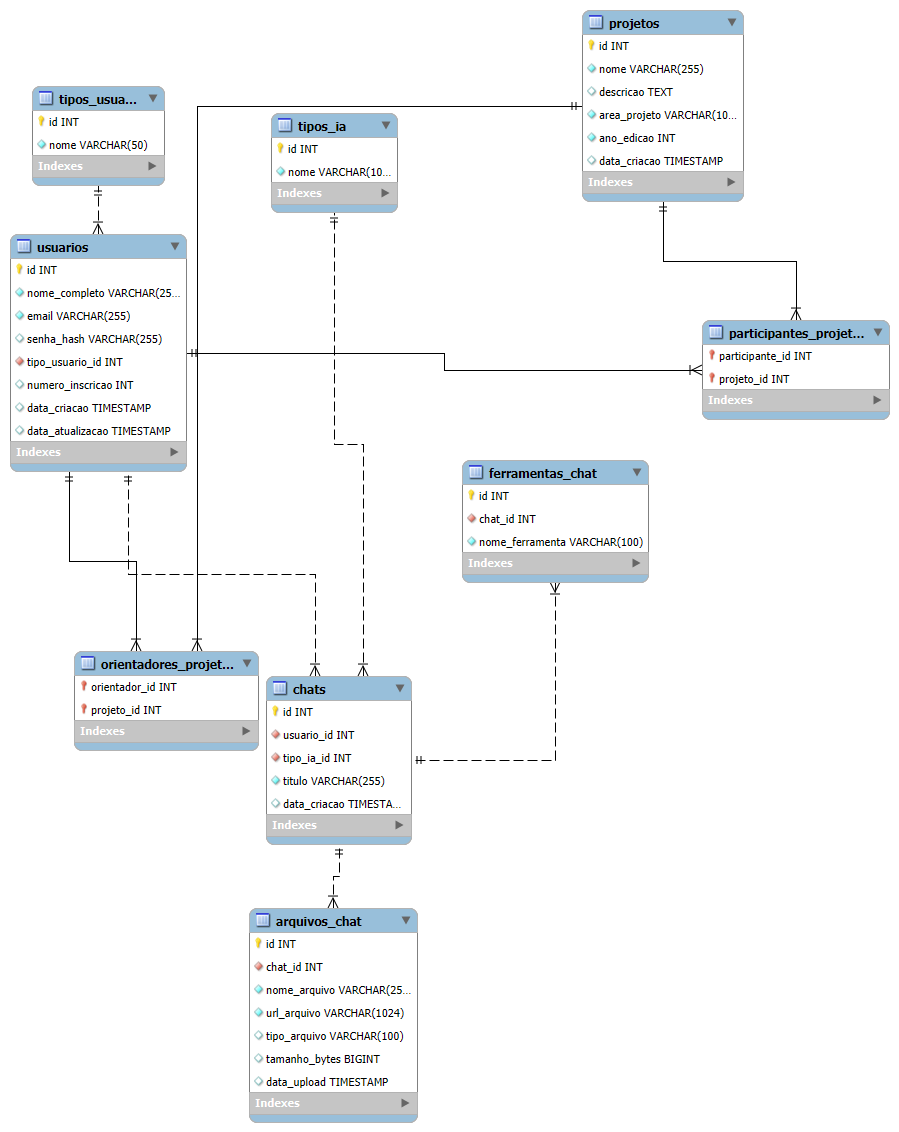
|  |
| --- |
| create table if not exists ferramentas\_chat ( -- ferramentas igual o chatgpt e gemini tem, nn é dificil fazer pois  -- a google documenta tudo id int auto\_increment primary key,  chat\_id int not null,  nome\_ferramenta varchar(100) not null, foreign key (chat\_id) references chats(id) on delete cascade, unique (chat\_id, nome\_ferramenta)  ); |

Print da estrutura da tabela ferramentas\_chat.



1. Por meio de engenharia reversa disponível no MySQL Workbench, gere o Modelo Físico de Dados, visando a representação das estruturas físicas de armazenamento de dados (tabelas) implementadas no SGBDR MySQL. Vale frisar que esse diagrama deve representar as chaves primárias e estrangeiras, bem como, as cardinalidades. (1,0 ponto)

[Modelo Físico de Dados gerado via engenharia reversa ou link de acesso à imagem aqui]

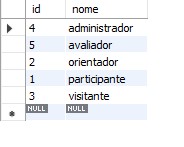


1. Insira 5 linhas (registros) em cada tabela e consulte tais registros após a inserção de todas as linhas. Cabe ressaltar que os scripts e prints devem estar legíveis e organizados. (1,0 ponto)

Script de inserção dos registros na tabela tipos\_usuario (copiar o script; não tirar print).

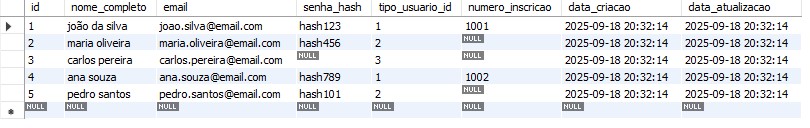
insert into tipos\_usuario (nome) values ('participante'), ('orientador'), ('visitante'), ('administrador'), ('avaliador');

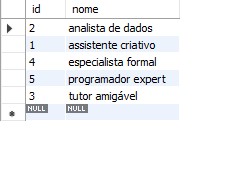
Print da consulta dos registros da tabela tipos\_usuario.



Script de inserção dos registros na tabela tipos\_ia (copiar o script; não tirar print).

insert into tipos\_ia (nome) values ('IA participante'), ('ia orientador'), ('ia visitante');

Print da consulta dos registros da tabela tipos\_ia.



Script de inserção dos registros na tabela usuarios (copiar o script; não tirar print).

insert into usuarios (nome\_completo, email, senha\_hash, tipo\_usuario\_id, numero\_inscricao) values

('joão da silva', 'joao.silva@email.com', 'hash123', 1, 1001),

('maria oliveira', 'maria.oliveira@email.com', 'hash456', 2, null),

('carlos pereira', 'carlos.pereira@email.com', null, 3, null),

('ana souza', 'ana.souza@email.com', 'hash789', 1, 1002),

('pedro santos', 'pedro.santos@email.com', 'hash101', 2, null);

Print da consulta dos registros da tabela usuarios.

Script de inserção dos registros na tabela projetos (copiar o script; não tirar print).

insert into projetos (nome, descricao, area\_projeto, ano\_edicao) values

('robô seguidor de linha', 'desenvolvimento de um robô autônomo', 'mecanica', 2024),

('análise de sentimentos em redes sociais', 'ia para analisar posts', 'informatia', 2024),

('sistema de irrigação automático', 'sistema iot para agricultura', 'mecanica', 2023),

('app de aprendizado de idiomas', 'aplicativo móvel para ensinar inglês', 'informatica', 2024),

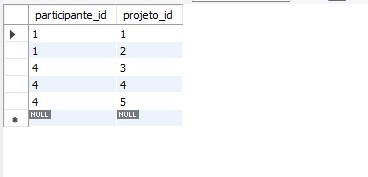
('circuito com arduino qualquer', 'algo que use arduino só pra ter eletronica aqui', 'eletronica', 2023);

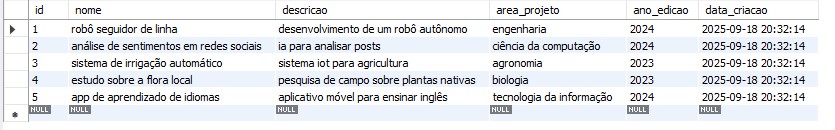
Print da consulta dos registros da tabela projetos.

Script de inserção dos registros na tabela participantes\_projetos (copiar o script; não tirar print).

|  |
| --- |
| insert into participantes\_projetos (participante\_id, projeto\_id) values  (1, 1),  (1, 2),  (4, 3),  (4, 4), (4, 5); |

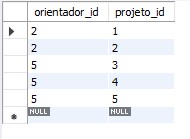
Print da consulta dos registros da tabela participantes\_projetos.



Script de inserção dos registros na tabela orientadores\_projetos (copiar o script; não tirar print).

|  |
| --- |
| insert into orientadores\_projetos (orientador\_id, projeto\_id) values  (2, 1),  (2, 2),  (5, 3),  (5, 4), (5, 5); |

Print da consulta dos registros da tabela orientadores\_projetos.



Script de inserção dos registros na tabela chats (copiar o script; não tirar print).

insert into chats (usuario\_id, tipo\_ia\_id, titulo) values

(1, 1, 'brainstorm para projeto robô'),

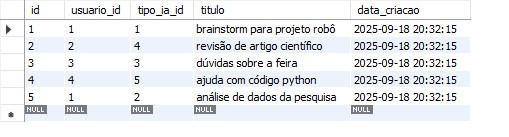
(2, 4, 'revisão de artigo científico'),

(3, 3, 'dúvidas sobre a feira'),

(4, 5, 'ajuda com código python'),

(1, 2, 'análise de dados da pesquisa');

Print da consulta dos registros da tabela chats.



Script de inserção dos registros na tabela arquivos\_chat (copiar o script; não tirar print).

insert into arquivos\_chat (chat\_id, nome\_arquivo, url\_arquivo, tipo\_arquivo, tamanho\_bytes) values

(1, 'esquema\_robo.pdf', 'http://files.com/esquema.pdf', 'application/pdf', 2048),

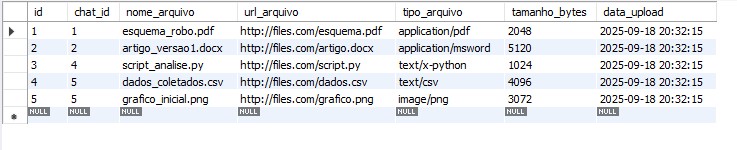
(2, 'artigo\_versao1.docx', 'http://files.com/artigo.docx', 'application/msword', 5120),

(4, 'script\_analise.py', 'http://files.com/script.py', 'text/x-python', 1024),

(5, 'dados\_coletados.csv', 'http://files.com/dados.csv', 'text/csv', 4096),

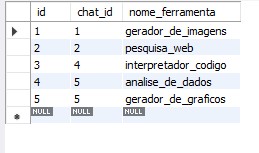
(5, 'grafico\_inicial.png', 'http://files.com/grafico.png', 'image/png', 3072);

Print da consulta dos registros da tabela arquivos\_chat.

Script de inserção dos registros na tabela ferramentas\_chat (copiar o script; não tirar print).

|  |
| --- |
| insert into ferramentas\_chat (chat\_id, nome\_ferramenta) values (1, 'gerador\_de\_imagens'),  (2, 'pesquisa\_web'),  (4, 'interpretador\_codigo'), (5, 'analise\_de\_dados'),  (5, 'gerador\_de\_graficos'); |

Print da consulta dos registros da tabela ferramentas\_chat.



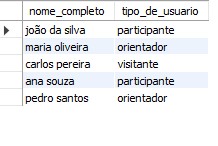
Obs: ta fora de ordem, execute as tabelas pai primeiro pra nn dar erro

1. Construa uma primeira query SQL para consultar as colunas (campos) de duas tabelas e uma segunda query para consultar as colunas (campos) de três tabelas. (1,0 ponto)

Script da primeira query (copiar o script; não tirar print).

select usuarios.nome\_completo, tipos\_usuario.nome as tipo\_de\_usuario from usuarios join tipos\_usuario on usuarios.tipo\_usuario\_id = tipos\_usuario.id;

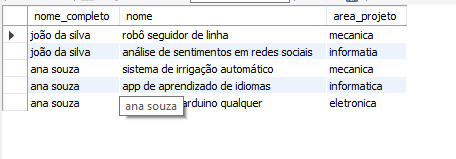
Print da primeira consulta.



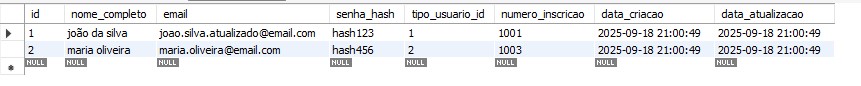
Script da segunda query (copiar o script; não tirar print).

select usuarios.nome\_completo, projetos.nome, projetos.area\_projeto from usuarios join participantes\_projetos on usuarios.id = participantes\_projetos.participante\_id join projetos on participantes\_projetos.projeto\_id = projetos.id;

Print da segunda consulta.



1. Atualize duas linhas (registros) de duas tabelas diferentes e consulte somente esses registros após a atualização. Cabe ressaltar que os scripts e prints devem estar legíveis e organizados. Nesta atividade, devem ser utilizados e exibidos os comandos begin e commit. (1,0 ponto)



Script de atualização dos dois registros da primeira tabela (copiar o script; não tirar print).

|  |
| --- |
| begin;    update usuarios  set email = 'joao.silva.atualizado@email.com' where id = 1;    update usuarios set numero\_inscricao = 1003  where id = 2;    commit; |

Print da consulta.

Script de atualização dos dois registros da segunda tabela (copiar o script; não tirar print).

|  |
| --- |
| begin;    update projetos set descricao = 'ia para analisar posts e comentários em redes sociais populares' where id = 2;    update projetos set ano\_edicao = 2024  where id = 3;    commit; |

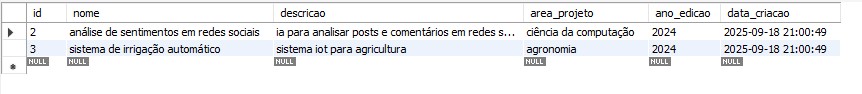
Print da consulta.

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

**Grupo 2: Apresentação do Banco de Dados do Projeto Integrador (3,0 pontos)**

Apresentar o Modelo e a Implementação do Banco de Dados do Projeto Integrador em aproximadamente 10 minutos, de acordo com a ordem estabelecida logo abaixo. Vale ressaltar que os artefatos produzidos nas atividades anteriores influenciam diretamente o conteúdo a ser apresentado. (3,0 pontos)

Ordem de Apresentação no dia 18/09:

13h50: Maria Catarina, Isabella, Mariane, Matheus

14h00: Leonardo, Nicolas, Vitor Moraes

14h10: Letícia, Sofia

14h20: Lucas, Marco Antonio, Vitor Hugo

14h30: Manuela, Luiz, Theo

14h40: Milena, Nicollas, Tamyres

14h50: Mirela, Pamela

15h30: Guilherme Salles, Gustavo, Julio

15h40: Arthur, João Carvalho, Laura

15h50: Daniel, Gabriel Henrique

16h00: Gabriel Prates, Guilherme Magioli

16h10: Amanda, Eloah, Gabriela

16h20: João Francisco, João Rocha, João Vilela

16h30: Emanuelly

16h40: Gabriel Ferreira