

ATIVIDADE– N4

GRADUAÇÃO: Tecnologia em Análise e Desenvolvimento de Sistemas

COORDENADOR: Prof. Me. Adriana Ohashi Kei Sato

DISCIPLINA: Desenvolvimento de Banco de Dados

DOCENTE: Prof. Márcio Cavalcante

DISCENTE: Felipe Schaitel

RM: 24112424

DESENVOLVIMENTO DE BANCO DE DADOS

PRIMEIRA PARTE DA ATIVIDADE IMPORTAR BD “faculdade” PARA XAMPP

1 BAIXAR NO MOODLE O ARQUIVO FACULDADE.sql

Neste atividade, você tem como objetivo demonstrar o que aprendeu



[20240502_orientacoes_atividade4.pptx](#)

[faculdade.sql](#)

Adicionar envio

2 ABRIR EDIÇÃO E COPIAR

```
Arquivo  Editar  Exibir  [Settings Icon]

-- Copiando estrutura para tabela faculdade.professor
CREATE TABLE IF NOT EXISTS `professor` (
  `id_pessoa` int(11) DEFAULT NULL,
  `email_profissional` varchar(100) DEFAULT NULL,
  `RF` varchar(10) DEFAULT NULL,
  `ativo` bit(1) DEFAULT NULL,
  UNIQUE KEY `id_pessoa` (`id_pessoa`),
  CONSTRAINT `FK_professor_pessoa` FOREIGN KEY (`id_pessoa`) REFERENCES `pessoa` (`id`) ON DELETE NO ACTION
ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

-- Copiando dados para a tabela faculdade.professor: ~2 rows (aproximadamente)
REPLACE INTO `professor` (`id_pessoa`, `email_profissional`, `RF`, `ativo`) VALUES
(1, 'marcio.cavalcante@belasartes.com.br', '111222334', NULL),
(3, 'joaodasilvateste@belasartes.com.br', '333333', NULL);

-- Copiando estrutura para tabela faculdade.usuario
CREATE TABLE IF NOT EXISTS `usuario` (
  `id_pessoa` int(11) DEFAULT NULL,
  `username` varchar(100) DEFAULT NULL,
  `senha` varchar(30) DEFAULT NULL,
  UNIQUE KEY `id_pessoa` (`id_pessoa`),
  CONSTRAINT `FK_usuario_pessoa` FOREIGN KEY (`id_pessoa`) REFERENCES `pessoa` (`id`) ON DELETE NO ACTION ON
UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_general_ci;

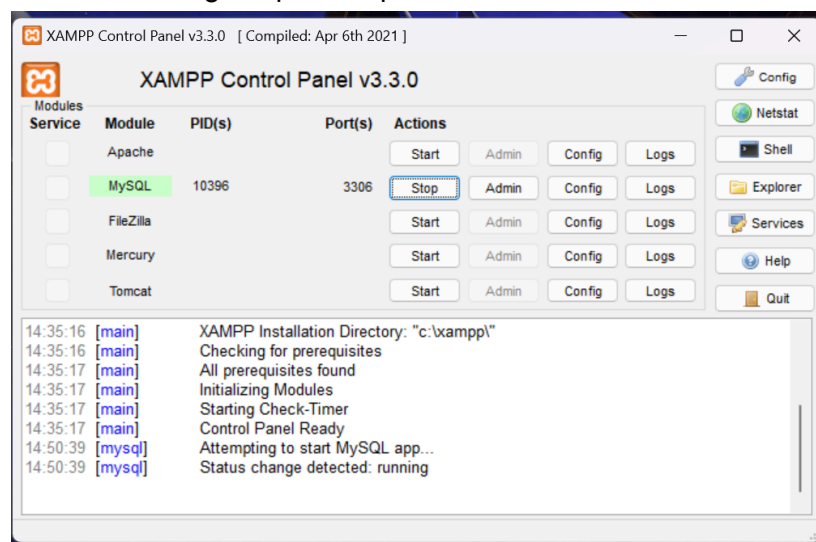
-- Copiando dados para a tabela faculdade.usuario: ~0 rows (aproximadamente)

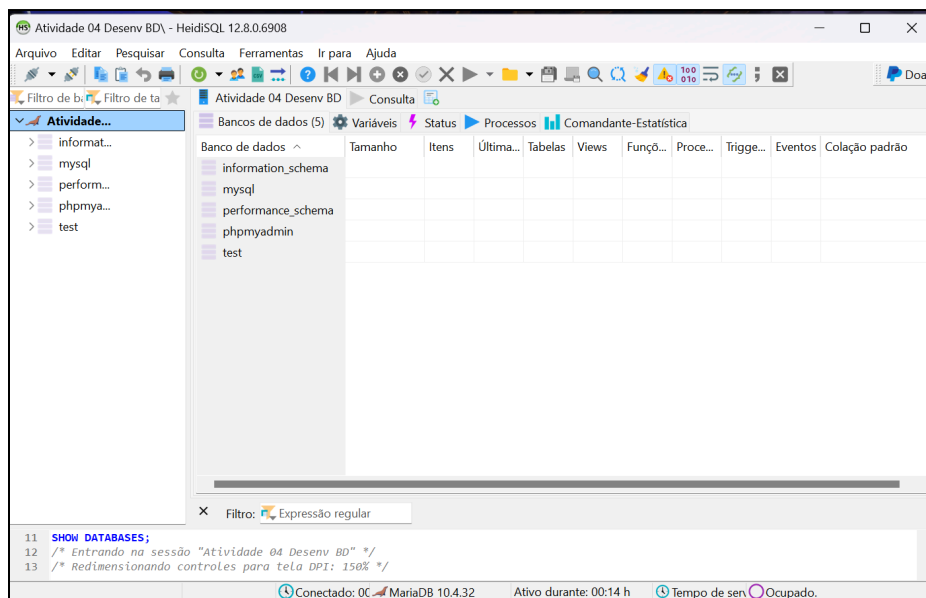
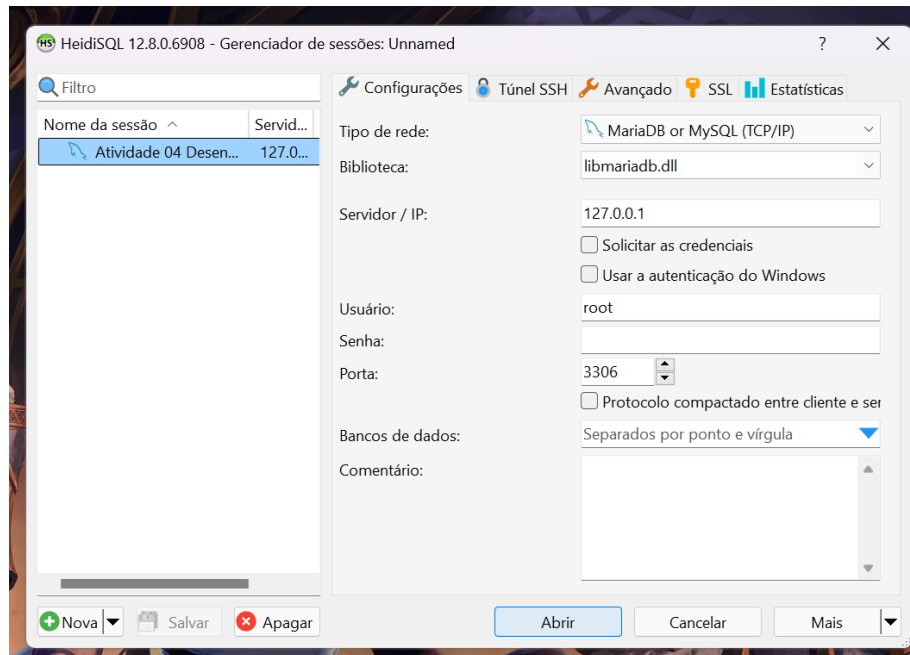
/*!40103 SET TIME_ZONE=IFNULL(@OLD_TIME_ZONE, 'system') */;
/*!40101 SET SQL_MODE=IFNULL(@OLD_SQL_MODE, '') */;
/*!40014 SET FOREIGN_KEY_CHECKS=IFNULL(@OLD_FOREIGN_KEY_CHECKS, 1) */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40111 SET SQL_NOTES=IFNULL(@OLD_SQL_NOTES, 1) */;

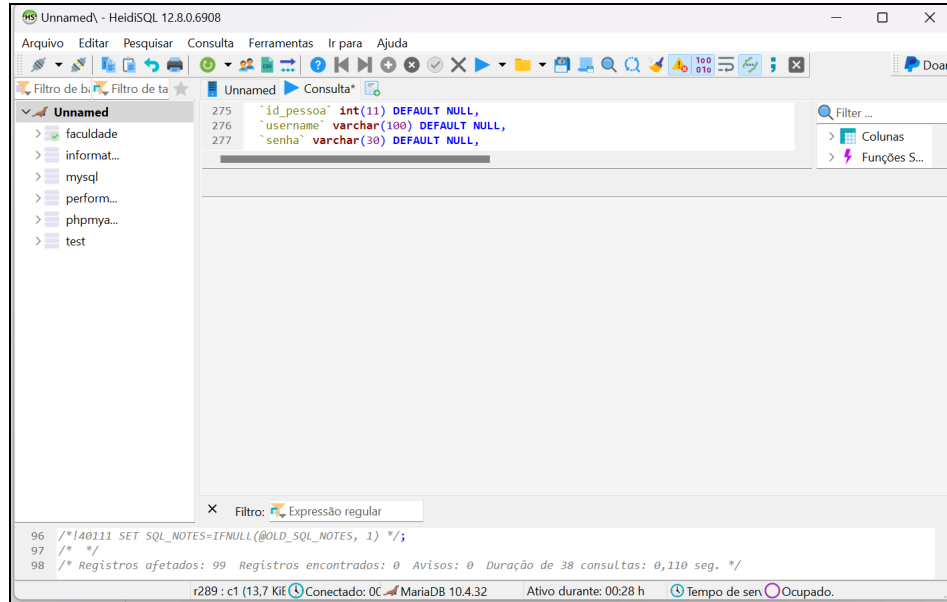
Ln 289, Col 1 | 13.787 de 13.787 caracteres | 100% | Windows (CRLF) | UTF-8
```

3 ABRIR XAMPP E CONECTAR A HEIDI.SQL

Clicar em consulta, colar o código copiado e pressionar executar:

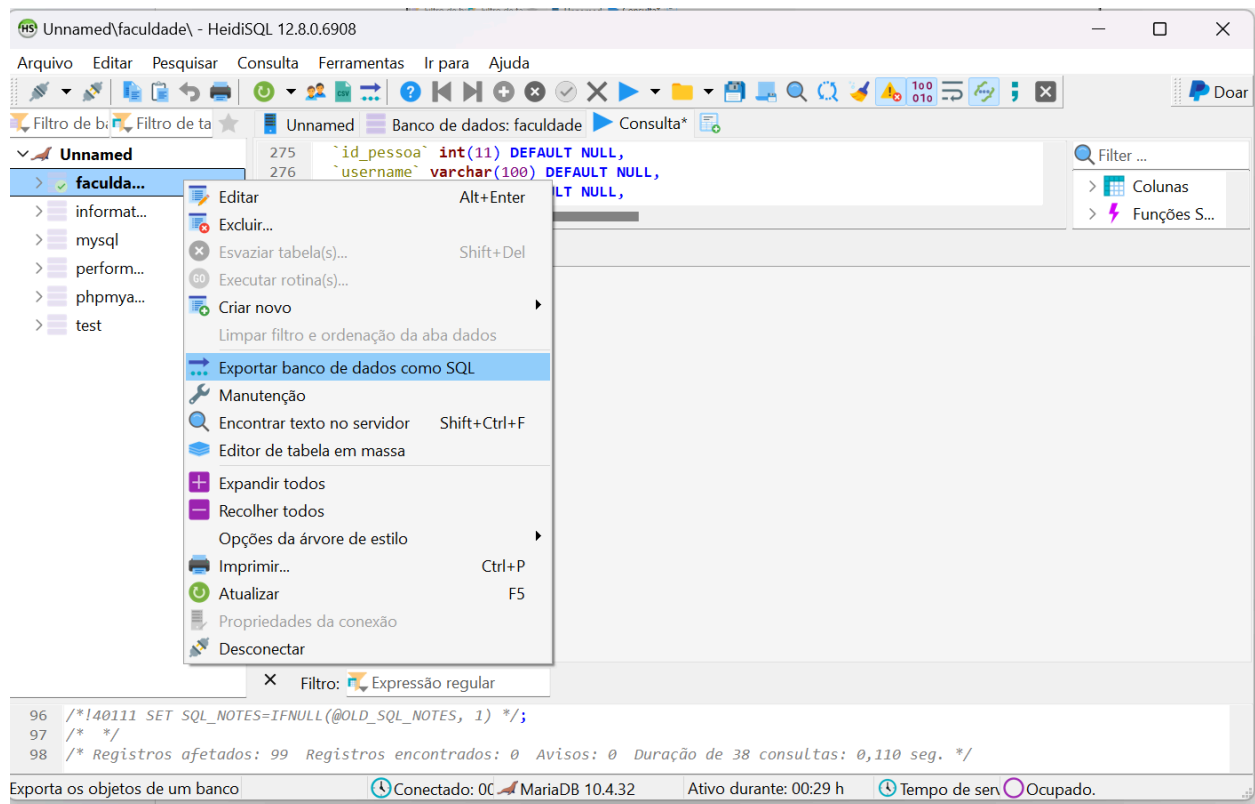


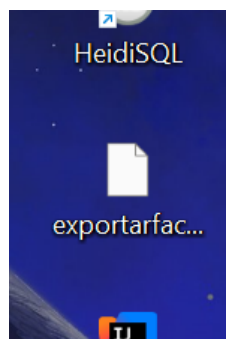
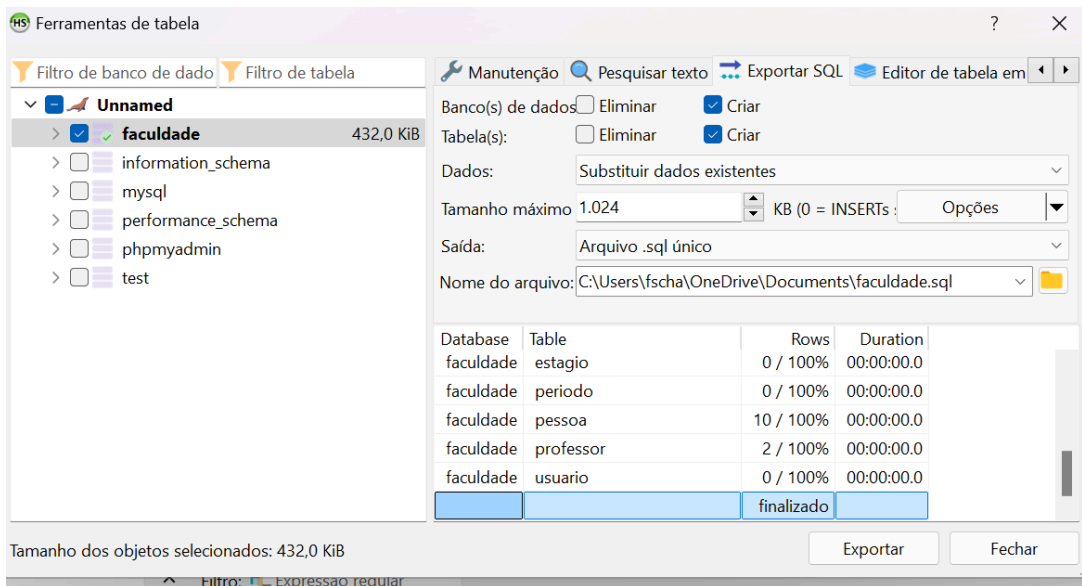
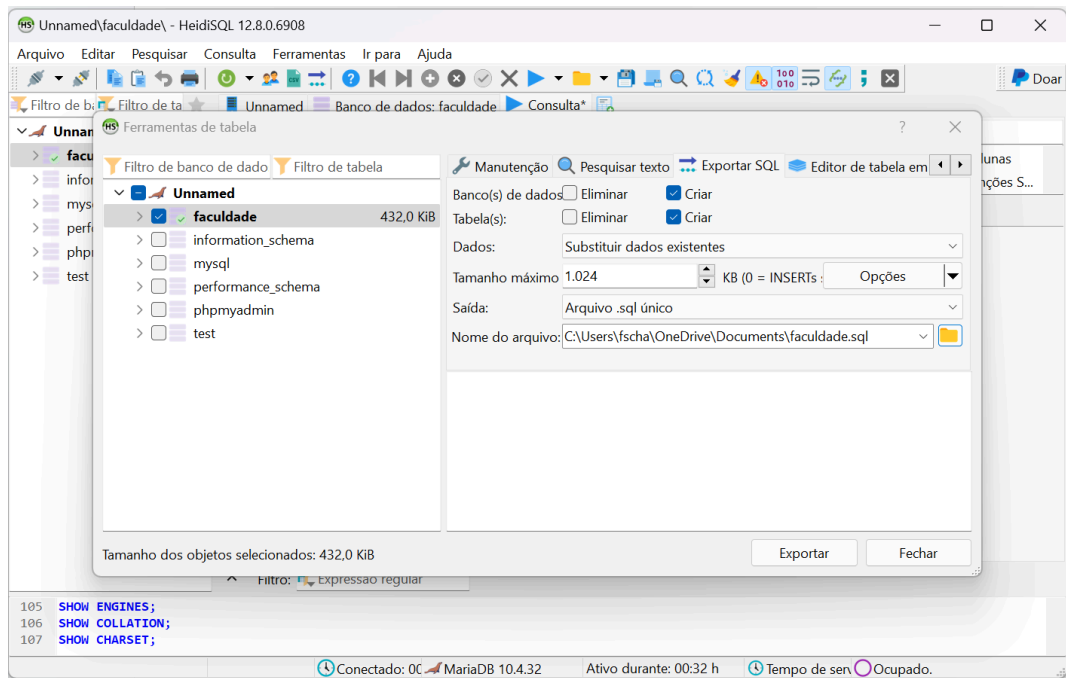




4 EXPORTAR BANCO DE DADOS “faculdade”

É necessário anexar o arquivo exportado ao Moodle para a entrega da atividade.





SEGUNDA PARTE DA ATIVIDADE - DER COM ENTIDADE “NOTAS”

Comparadores =, >, <, >= IN ou NOT IN

Conceito JOIN

Comando UPDATE: SELECT, FROM, WHERE, GROUP BY, HAVING SUM

CREATE DATABASE = cria o banco de dados

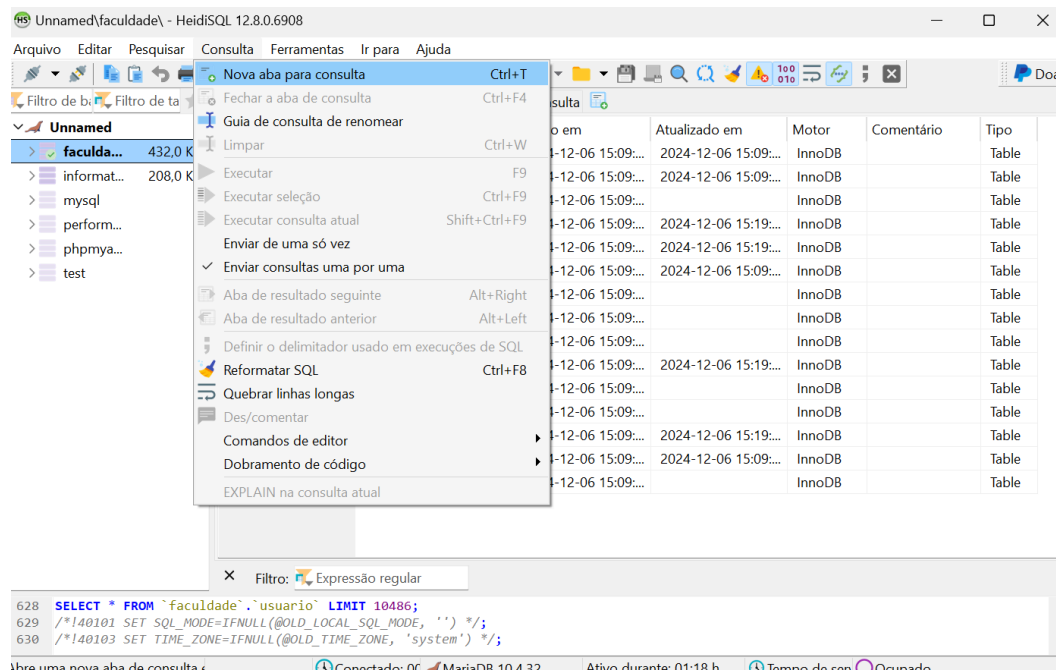
Id = identificador dos nomes dos alunos é a chave primária

Id = chave primária para as disciplinas

Id = chave primária da tabela

Aluno_id relaciona aluno e disciplina e Disciplina_id relaciona disciplina com aluno.

CASE = para calcular a media. Se a media da nota for NULL tem que retornar como “SEM NOTA”



Unnamed\faculdade\ - HeidiSQL 12.8.0.6908

Arquivo Editar Pesquisar Consulta Ferramentas Ir para Ajuda

Filtro de banco de dados Filtro de tabela

Unnamed Banco de dados: faculdade Consulta*

Unamed

- faculda... 432,0 KiB
- informat... 208,0 KiB
- mysql
- perform...
- phpmya...
- test

```

73 END AS media
74 FROM
75 alunos
76 LEFT JOIN
77 disciplinas_realizadas ON alunos.id = disciplinas_realizadas.aluno_id
78 LEFT JOIN
79 disciplinas ON disciplinas_realizadas.disciplina_id = disciplinas.id
80 LEFT JOIN
81 notas ON alunos.id = notas.aluno_id AND disciplinas.id = notas.disciplina_id
82 GROUP BY
83 alunos.id, disciplinas.id
84 ORDER BY
85 alunos.nome, disciplinas.nome;
86
87

```

Resultado #1 (5r x 3c)

#	aluno	disciplina	media
1	Ana Costa	Matemática	SEM NOTA
2	João Silva	Física	8.000000
3	João Silva	Matemática	7.500000
4	Maria Oliveira	Química	6.500000
5	Pedro Santos	História	9.000000

Filtro: Expressão regular

```

675 -- Consultando os alunos e suas médias SELECT alunos.nome AS aluno, disciplinas.nome AS disciplina, CASE
676 /* */
677 /* Registros afetados: 18 Registros encontrados: 5 Avisos: 0 Duração de 11 consultas: 0,063 seg. */

```

CÓDIGO

```

CREATE DATABASE IF NOT EXISTS faculdade_simplificada;
USE faculdade_simplificada;

```

```

CREATE TABLE alunos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100) NOT NULL
);

```

```

INSERT INTO alunos (nome) VALUES
    ('João Silva'),
    ('Maria Oliveira'),
    ('Pedro Santos'),
    ('Ana Costa');

```

```
CREATE TABLE disciplinas (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nome VARCHAR(100) NOT NULL  
);
```

```
INSERT INTO disciplinas (nome) VALUES  
    ('Matemática'),  
    ('Física'),  
    ('Química'),  
    ('História');
```

```
CREATE TABLE disciplinas_realizadas (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    aluno_id INT,  
    disciplina_id INT,  
    concluiu BOOLEAN DEFAULT FALSE,  
    FOREIGN KEY (aluno_id) REFERENCES alunos(id),  
    FOREIGN KEY (disciplina_id) REFERENCES disciplinas(id)  
);
```

```
INSERT INTO disciplinas_realizadas (aluno_id, disciplina_id,  
concluiu) VALUES  
    (1, 1, TRUE),  
    (1, 2, TRUE),  
    (2, 3, FALSE),  
    (3, 4, TRUE),  
    (4, 1, TRUE);
```

```
CREATE TABLE notas (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    aluno_id INT,  
    disciplina_id INT,  
    nota DECIMAL(5, 2),  
    FOREIGN KEY (aluno_id) REFERENCES alunos(id),  
    FOREIGN KEY (disciplina_id) REFERENCES disciplinas(id)  
);
```

```
INSERT INTO notas (aluno_id, disciplina_id, nota) VALUES  
    (1, 1, 7.5),  
    (1, 2, 8.0),
```



```

(2, 3, 6.5),
(3, 4, 9.0);

SELECT
    alunos.nome AS aluno,
    disciplinas.nome AS disciplina,
    CASE
        WHEN AVG(notas.nota) IS NULL THEN 'SEM NOTA'
        ELSE AVG(notas.nota)
    END AS media
FROM
    alunos
LEFT JOIN
    disciplinas_realizadas ON alunos.id =
    disciplinas_realizadas.aluno_id
LEFT JOIN
    disciplinas ON disciplinas_realizadas.disciplina_id =
    disciplinas.id
LEFT JOIN
    notas ON alunos.id = notas.aluno_id AND disciplinas.id =
    notas.disciplina_id
GROUP BY
    alunos.id, disciplinas.id
ORDER BY
    alunos.nome, disciplinas.nome;

```

Referências bibliográficas

Material didático da disciplina. Disponível no Moodle BA

W3Schools. *SQL Tutorial*. Disponível em: <https://www.w3schools.com/sql/>. Acesso em: 20 nov. 2024.

W3Schools. *SQL CASE Statement*. Disponível em: https://www.w3schools.com/sql/sql_case.asp. Acesso em: 20 nov. 2024.

Microsoft. *SQL Samples - Where Are They?*. Disponível em: <https://learn.microsoft.com/pt-br/sql/samples/sql-samples-where-are?view=sql-server-ver16>. Acesso em: 20 nov. 2024.

W3Schools. *SQL Quick Reference*. Disponível em: https://www.w3schools.com/sql/sql_quickref.asp. Acesso em: 20 nov. 2024.

W3Schools. *SQL Examples*. Disponível em: https://www.w3schools.com/sql/sql_examples.asp. Acesso em: 20 nov. 2024.