BANCO DE DADOS I
TRABALHO EM GRUPO – ATÉ 4 INTEGRANTES

Nome do(s) Aluno(s):
1. **Arthur Pimentel**
2. **Arthur Tomazelli**
3. **Eduardo Alcaria**
4. **Eduarda Leguisamo**

Descrição do banco de dados

Considerando o seguinte banco de dados:

PROFESSOR	**numeroProf**, nome, idade, salário, departamento
ALUNO	**matriuculaAluno**, nome, idade, endereço, status
DISCIPLINA	**codigoDisc**, nome, departamento, cargaHoraria
TURMA	**codigoDisc, semestre,** numeroProf
HISTÓRICO	**matriculaAluno, codigoDisc, semestre**, nota

Chaves:

Tabela	Chave Primária	Chave Estrangeira
PROFESSOR	**numeroProf**	
ALUNO	**matriculaAluno**	
DISCIPLINA	**codigoDisc**	
TURMA	**codigoDisc, semestre**	numeroProf REFERENCIA PROFESSOR codigoDisc REFERENCIA DISCIPLINA
HISTÓRICO	**matriculaAluno, codigoDisc, semestre**	matriculaAluno REFERENCIA ALUNO (codigoDisc, semestre) REFERENCIA TURMA

Observações:

- Tabela **PROFESSOR**:
  - idade tem valor máximo de 70.
- Tabela **ALUNO**
  - status representa a situação do vínculo do aluno com a Universidade; os valores possíveis são ‘matriculado’, ‘não matriculado’ ou ‘evadido’.
- Tabela **DISCIPLINA**:
  - cargaHoraria é carga horária da disciplina; os valores possíveis são 30, 60, ou 120.
  - o valor mais comum para carga horária é 60.
- Tabela **HISTÓRICO**:
  - nota somente pode ter valores entre 0 e 10.

É importante definir algumas **restrições de integridade** para que os dados inseridos respeitem às regras de validação e, consequentemente, o banco de dados permaneça sempre íntegro,

com os dados corretos segundo o seu contexto de aplicação.

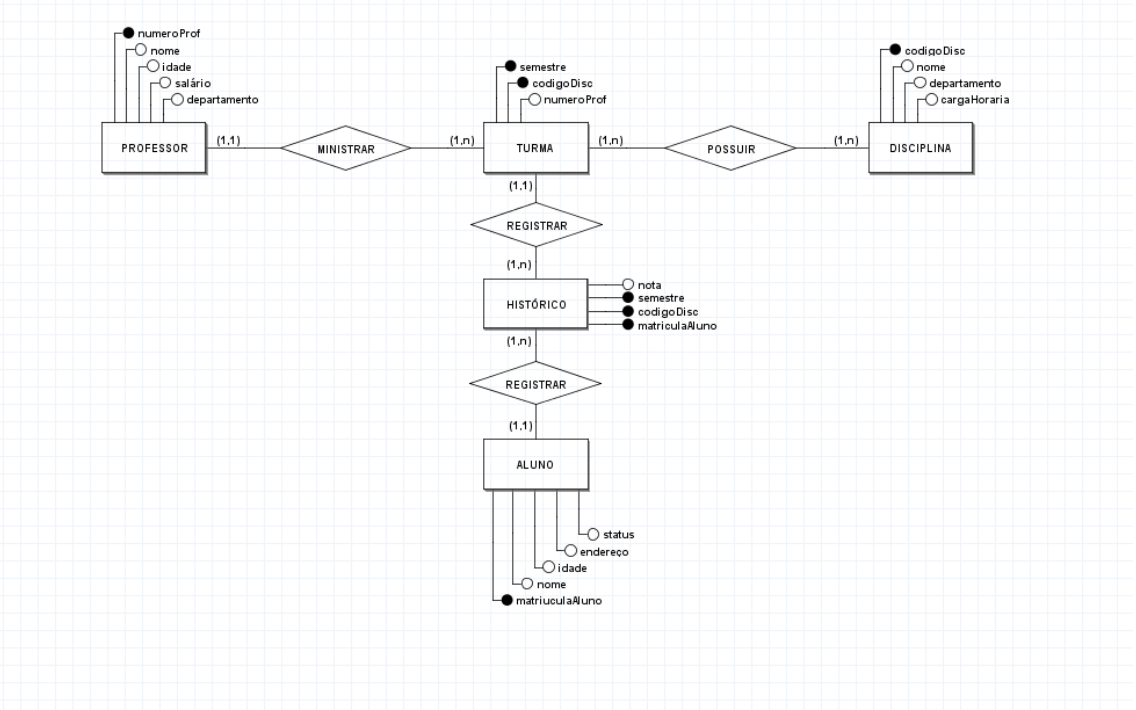
Conceito de restrição de integridade: são regras definidas sobre os dados para regular as operações que podem ser realizadas sobre eles.

Além das restrições, é possível definir valores ***default*** para serem inseridos em alguns atributos (colunas) determinando um valor que será inserido nessa coluna da tabela, quando um valor não for enviado pela instrução INSERT.

Considere o exposto acima para criar o banco de dados neste trabalho.

Questões do trabalho:

1. (1 PONTO) Apresente, abaixo, o DER que representa o projeto do banco de dados dado no trabalho. Utilize a notação apresentada em aula e a ferramenta brModelo para construir o DER.



2. (2 PONTOS) Crie o banco de dados descrito acima

   Os elementos listados abaixo não podem faltar na criação das tabelas do banco de dados:

   2.1. Defina as **chaves primárias** apropriadas, utilize as cláusulas `CONSTRAINT` e `PRIMARY KEY` para isso.

   2.2. Defina as **chaves estrangeiras** apropriadas, utilize as cláusulas `CONSTRAINT` e `FOREIGN KEY` para isso.

2.3. Defina algumas **restrições de integridade** para o banco de dados; as restrições estão sugeridas no detalhamento do projeto do banco de dados descrito acima.

As restrições podem ser criadas através de uma `CONSTRAINT` da seguinte forma:

```
CONSTRAINT <nome da constraint> CHECK (<expressão>)
```

onde `<expressão>` é alguma expressão envolvendo o atributo relacionado na restrição. Ela pode envolver os operadores `IN` e `BETWEEN`.

Exemplos:

```
CONSTRAINT ck_idade CHECK (idade > 18)
ou
CONSTRAINT ck_ramo CHECK (ramo IN ('Atacado', 'Varejo', 'Serviços'))
```

2.4. Defina alguns valores ***default***; os valores *default* estão sugeridos no detalhamento do projeto do banco de dados descrito acima.

Valores default podem ser criados da seguinte forma:

```
DEFAULT <valor>, ao lado da definição do atributo na criação da tabela.
```

Exemplo para a tabela EMPREGADO:

```
…
idade number DEFAULT 60,
…
```

Observação: para ver a descrição completa da tabela, após tê-la criado no ORACLE, execute a instrução:

```
DESCRIBE <nome da tabela>;
```

*** sugestão: copie e cole o seguinte conjunto de instruções em um arquivo texto e grave em seu computador, assim será possível criar um script para execução de todas as instruções em conjunto no Oracle ***
*** o conjunto de instruções DROP TABLE abaixo remove as tabelas do banco de dados ***
*** a diretiva CASCADE CONSTRAINTS serve para que as tabelas possam ser apagadas, independentemente se chaves estrangerias de outras tabelas fazem referencia a elas ***

```
DROP TABLE PROFESSOR CASCADE CONSTRAINTS;
DROP TABLE ALUNO CASCADE CONSTRAINTS;
DROP TABLE DISCIPLINA CASCADE CONSTRAINTS;
DROP TABLE TURMA CASCADE CONSTRAINTS;
DROP TABLE HISTORICO CASCADE CONSTRAINTS;
```

*** sugestão: copie e cole o seguinte conjunto de instruções em um arquivo texto e grave em seu computador, assim será possível criar um script para execução de todas as instruções em conjunto no Oracle ***
*** o conjunto de instruções CREATE TABLE cria as tabelas descritas acima no banco de dados
ENTRETANTO AS INSTRUÇÕES ESTÃO INCOMPLETAWS E FAZ PARTE DO TRABALHO COMPLETÁ-LAS CONFORME O QUE ESTÁ DESCRITO NO INÍCIO DESTE ENUNCIADO
***

```
CREATE TABLE PROFESSOR (
numeroProf NUMBER,
nome VARCHAR2(15),
idade NUMBER,
salario NUMBER,
```

```
departamento VARCHAR2(15));

CREATE TABLE ALUNO (
matriculaAluno NUMBER PRIMARY KEY,
nome VARCHAR2(15),
idade NUMBER,
endereco VARCHAR2(15),
status VARCHAR2(30));

CREATE TABLE DISCIPLINA (
codigoDisc NUMBER PRIMARY KEY,
nome VARCHAR2(15),
departamento VARCHAR2(15),
cargaHoraria NUMBER DEFAULT 60);

CREATE TABLE TURMA (
codigoDisc NUMBER,
semestre VARCHAR2(5),
numeroProf NUMBER,
CONSTRAINT fk_TURMA_DISCIPLINA FOREIGN KEY (codigoDisc) REFERENCES
DISCIPLINA(codigoDisc));

CREATE TABLE HISTORICO (
matriculaAluno NUMBER,
codigoDisc NUMBER,
semestre VARCHAR2(5),
nota NUMBER,
CONSTRAINT pk_HISTORICO PRIMARY KEY (matriculaAluno, codigoDisc, semestre),
CONSTRAINT fk_HISTORICO_TURMA FOREIGN KEY (codigoDIsc, semestre) REFERENCES
TURMA(codigoDisc, semestre));
```

```
CREATE TABLE DISCIPLINA (
  codigoDisc    NUMBER,
  nome          VARCHAR2(100),
  departamento  VARCHAR2(50),
  cargaHoraria  NUMBER DEFAULT 60,
  CONSTRAINT pk_disciplina PRIMARY KEY (codigoDisc),
  CONSTRAINT ck_disciplina_carga CHECK (cargaHoraria IN (30,60,120))
);


CREATE TABLE PROFESSOR (
  numeroProf   NUMBER,
  nome         VARCHAR2(100),
  idade        NUMBER,
  salario      NUMBER,
  departamento VARCHAR2(50),
  CONSTRAINT pk_professor PRIMARY KEY (numeroProf),
  CONSTRAINT ck_professor_idade CHECK (idade BETWEEN 0 AND 70)
);


CREATE TABLE ALUNO (
  matriculaAluno NUMBER,
  nome           VARCHAR2(100),
  idade          NUMBER,
  endereco       VARCHAR2(200),
  status         VARCHAR2(30) DEFAULT 'matriculado',
  CONSTRAINT pk_aluno PRIMARY KEY (matriculaAluno),
  CONSTRAINT ck_aluno_status CHECK (status IN ('matriculado','não matriculado','evadido'))
);


CREATE TABLE TURMA (
```

```
  codigoDisc NUMBER,
  semestre   VARCHAR2(10),
  numeroProf NUMBER,
  CONSTRAINT pk_turma PRIMARY KEY (codigoDisc, semestre),
  CONSTRAINT fk_turma_disciplina FOREIGN KEY (codigoDisc)
    REFERENCES DISCIPLINA (codigoDisc),
  CONSTRAINT fk_turma_professor FOREIGN KEY (numeroProf)
    REFERENCES PROFESSOR (numeroProf)
);

CREATE TABLE HISTORICO (
  matriculaAluno NUMBER,
  codigoDisc     NUMBER,
  semestre       VARCHAR2(10),
  nota           NUMBER,
  CONSTRAINT pk_historico PRIMARY KEY (matriculaAluno, codigoDisc, semestre),
  CONSTRAINT fk_historico_aluno FOREIGN KEY (matriculaAluno)
    REFERENCES ALUNO (matriculaAluno),
  CONSTRAINT fk_historico_turma FOREIGN KEY (codigoDisc, semestre)
    REFERENCES TURMA (codigoDisc, semestre),
  CONSTRAINT ck_historico_nota CHECK (nota BETWEEN 0 AND 10)
);
```

3. (1 PONTOS) Popule as tabelas do banco de dados, inserindo algumas (poucas) tuplas em cada tabela. Não esqueça de respeitar todas as restrições definidas na criação das tabelas. **Faça a inserção de algumas tuplas (MÁXIMO 5 tuplas por tabela), conforme observações abaixo.**

   Observação 1: para omitir alguns valores em uma instrução `SQL INSERT`, indique, antes da palavra `VALUES`, os nomes dos atributos para os quais vai inserir valores.

   Exemplo:

   ```
   INSERT INTO EMPREGADO (Cod_Emp, Nome_Emp) values (100, 'Denise Bandeira');
   ```

   No exemplo acima, será inserido um empregado com apenas os valores para os atributos Cod_Emp e Nome_Emp. Os demais atributos ficarão com valores “nulos”, a não ser o atributo idade, que ficará com o valor *default* (caso tenha sido definido na criação da tabela).

   Observação 2: Insira pelo menos uma linha no banco de dados onde o valor Default seja aplicado.

   Observação 3: Insira 3 linhas (ao todo), que gerem erros de cada um dos tipos listados abaixo:
   - 1 Erro de Integridade Referencial
   - 1 Erro de Integridade de Entidade
   - 1 Erro de validação de restrição de integridade do tipo CHECK

Inserts Válidos

PROFESSOR

INSERT INTO PROFESSOR (numeroProf, nome, idade, salario, departamento) VALUES (1, 'Ana Silva', 34, 7500, 'Comp. Sci');

INSERT INTO PROFESSOR (numeroProf, nome, idade, salario, departamento) VALUES (2, 'Bruno Costa', 24, 6800, 'Matematica');

INSERT INTO PROFESSOR (numeroProf, nome, idade, salario, departamento) VALUES (3, 'Clara Souza', 67, 9000, 'Física');

INSERT INTO PROFESSOR (numeroProf, nome, idade, salario, departamento) VALUES (4, 'Daniel Reis', 70, 12000, 'Comp. Sci');

LISTAGEM

	NUMEROPROF	NOME	IDADE	SALARIO	DEPARTAMENTO
1	1	Ana Silva	34	7500	Comp. Sci
2	2	Bruno Costa	24	6800	Matematica
3	3	Clara Souza	67	9000	Física
4	4	Daniel Reis	70	12000	Comp. Sci

DISCIPLINA

INSERT INTO DISCIPLINA (codigoDisc, nome, departamento, cargaHoraria) VALUES (10001, 'Programação I', 'Comp. Sci', 60);

INSERT INTO DISCIPLINA (codigoDisc, nome, departamento) VALUES (10002, 'Matemática Discreta', 'Matematica');
O INSERT acima NÃO fornece cargaHoraria -> será aplicado o DEFAULT 60

INSERT INTO DISCIPLINA (codigoDisc, nome, departamento, cargaHoraria) VALUES (10003, 'Física Básica', 'Física', 30);

INSERT INTO DISCIPLINA (codigoDisc, nome, departamento, cargaHoraria) VALUES (10004, 'Algoritmos', 'Comp. Sci', 120);

LISTAGEM

	CODIGODISC	NOME	DEPARTAMENTO	CARGAHORARIA
1	10001	Programação I	Comp. Sci	60
2	10002	Matemática Discreta	Matematica	60
3	10003	Física Básica	Física	30
4	10004	Algoritmos	Comp. Sci	120

ALUNO

INSERT INTO ALUNO (matriculaAluno, nome, idade, endereco, status) VALUES (1001, 'Joao Pereira', 20, 'Rua A, 123', 'matriculado');

INSERT INTO ALUNO (matriculaAluno, nome, idade, endereco) VALUES (1002, 'Maria Lima', 22, 'Av. B, 45');
O INSERT acima NÃO fornece status -> será aplicado DEFAULT 'matriculado'

INSERT INTO ALUNO (matriculaAluno, nome, idade, endereco, status) VALUES (1003, 'Pedro Almeida', 23, 'Rua C, 9', 'não matriculado');

LISTAGEM

	MATRICULAALUNO	NOME	IDADE	ENDERECO	STATUS
1	1001	Joao Pereira	20	Rua A, 123	matriculado
2	1002	Maria Lima	22	Av. B, 45	matriculado
3	1003	Pedro Almeida	23	Rua C, 9	não matriculado

TURMA

INSERT INTO TURMA (codigoDisc, semestre, numeroProf) VALUES (10001, '20252', 1);
INSERT INTO TURMA (codigoDisc, semestre, numeroProf) VALUES (10002, '20252', 2);
INSERT INTO TURMA (codigoDisc, semestre, numeroProf) VALUES (10003, '20251', 3);
INSERT INTO TURMA (codigoDisc, semestre, numeroProf) VALUES (10004, '20252', 4);

LISTAGEM

	CODIGODISC	SEMESTRE	NUMEROPROF
1	10001	20252	1
2	10002	20252	2
3	10003	20251	3
4	10004	20252	4

HISTÓRICO

INSERT INTO HISTORICO (matriculaAluno, codigoDisc, semestre, nota) VALUES (1001, 10001, '20252', 8.5);

INSERT INTO HISTORICO (matriculaAluno, codigoDisc, semestre, nota) VALUES (1001, 10002, '20252', 6.0);

INSERT INTO HISTORICO (matriculaAluno, codigoDisc, semestre, nota) VALUES (1002, 10001, '20252', 7.5);

INSERT INTO HISTORICO (matriculaAluno, codigoDisc, semestre, nota) VALUES (1003, 10003, '20251', 9.0);

LISTAGEM

	MATRICULAALUNO	CODIGODISC	SEMESTRE	NOTA
1	1001	10001	20252	8.5
2	1001	10002	20252	6
3	1002	10001	20252	7.5
4	1003	10003	20251	9

ERROS

-- ERRO - Integridade Referencial (inserir TURMA com codigoDisc que NÃO existe)
-- ERRO esperado: violação de FK (fk_turma_disciplina)
INSERT INTO TURMA (codigoDisc, semestre, numeroProf) VALUES (99999, '20252', 1);

```
SQL> INSERT INTO TURMA (codigoDisc, semestre, numeroProf) VALUES (99999, '20252', 1)

ORA-02291: integrity constraint (EDUARDOALCARIALOP_SCHEMA_DCX9I.FK_TURMA_DISCIPLINA) violated - parent key not found

https://docs.oracle.com/error-help/db/ora-02291/
Error at Line: 4 Column: 0
```

-- ERRO - Integridade de Entidade (inserir ALUNO com PK nula ou duplicada)
-- ERRO esperado: violação de PK (pk_aluno) se matriculaAluno já existir; ou de NOT NULL se a PK fosse nula
-- (ex.: duplicata de 1001)
INSERT INTO ALUNO (matriculaAluno, nome, idade, endereco) VALUES (1001, 'Duplicado', 21, 'Rua X');

```
SQL> INSERT INTO ALUNO (matriculaAluno, nome, idade, endereco) VALUES (1001, 'Duplicado', 21, 'Rua X')

ORA-00001: unique constraint (EDUARDOALCARIALOP_SCHEMA_DCX9I.PK_ALUNO) violated on table EDUARDOALCARIALOP_SCHEMA_DCX9I.ALUNO columns (MATRICULAALUNO)
ORA-03301: (ORA-00001 details) row with column values (MATRICULAALUNO:1001) already exists

https://docs.oracle.com/error-help/db/ora-00001/
Error at Line: 4 Column: 0
```

-- ERRO - Validação CHECK (idade ou carga fora do permitido)
-- ERRO esperado: violação de CHECK (ck_professor_idade)
INSERT INTO PROFESSOR (numeroProf, nome, idade, salario, departamento) VALUES (5, 'Idade Errada', 80, 5000, 'Teste');

```
SQL> INSERT INTO PROFESSOR (numeroProf, nome, idade, salario, departamento) VALUES (5, 'Idade Errada', 80, 5000, 'Teste')

ORA-02290: check constraint (EDUARDOALCARIALOP_SCHEMA_DCX9I.CK_PROFESSOR_IDADE) violated

https://docs.oracle.com/error-help/db/ora-02290/
Error at Line: 4 Column: 0
```

4. (6 PONTOS) Realize as seguintes consultas em SQL:

4.1. Listar o nome dos professores que tenham idade menor do que 25 anos e que ministraram disciplinas em 20252;

SQL utilizada:

```
SELECT Professor.Nome
FROM Professor INNER JOIN Turma ON Professor.NumeroProf = Turma.NumeroProf
WHERE Professor.Idade < 25
AND Turma.Semestre = 20252;
```

Resultado:

```
NOME
-------------
Bruno Costa

Elapsed: 00:00:00.005
1 rows selected.
```

	NOME
1	Bruno Costa

4.2. Listar o nome dos professores, em ordem alfabética, que ministraram a disciplina de código 10001.

SQL utilizada:

```
SELECT Professor.Nome
FROM Professor INNER JOIN Turma ON Professor.NumeroProf = Turma.NumeroProf
WHERE Turma.CodigoDisc = 10001
ORDER BY Professor.Nome;
```

Resultado:

```
NOME
-----------
Ana Silva

Elapsed: 00:00:00.007
1 rows selected.
```

	NOME
1	Ana Silva

4.3. Quantidade de disciplinas com carga horária > 60 que cada professor com idade > 65 já ministrou.

SQL utilizada:

```
SELECT Professor.Nome, COUNT(*) AS Qnt_Disciplinas
FROM Professor INNER JOIN Turma ON Professor.NumeroProf = Turma.NumeroProf
        INNER JOIN Disciplina ON Turma.CodigoDisc = Disciplina.CodigoDisc
WHERE Disciplina.CargaHoraria > 60
AND Professor.Idade > 65
GROUP BY Professor.Nome;
```

Resultado:

```
NOME         QNT_DISCIPLINAS
------------ ---------------
Daniel Reis   1

Elapsed: 00:00:00.024
1 rows selected.
```

	NOME	QNT_DISCIPLINAS
1	Daniel Reis	1

4.4. Listar matrícula, nome do aluno, nome das disciplinas cursadas e notas, somente quando nota ≥ 7.

SQL utilizada:

```
SELECT Aluno.MatriculaAluno AS Matrícula, Aluno.Nome, Disciplina.Nome AS Disciplina,
Historico.Nota
FROM Aluno INNER JOIN Historico ON Aluno.MatriculaAluno = Historico.MatriculaAluno
        INNER JOIN Disciplina ON Historico.CodigoDisc = Disciplina.CodigoDisc
WHERE Historico.Nota >= 7;
```

Resultado:

```
MATRÍCULA NOME           DISCIPLINA      NOTA
--------- -------------- --------------- ----
1001      Joao Pereira    Programação I   8.5
1002      Maria Lima      Programação I   7.5
1003      Pedro Almeida   Física Básica   9
```

Elapsed: 00:00:00.007
3 rows selected.

	MATRÍCULA	NOME	DISCIPLINA	NOTA
1	1001	Joao Pereira	Programação I	8.5
2	1002	Maria Lima	Programação I	7.5
3	1003	Pedro Almeida	Física Básica	9

4.5. Média geral das notas de cada aluno.

SQL utilizada:

```
SELECT Aluno.MatriculaAluno AS Matrícula, AVG(Historico.Nota) AS Média_Notas
FROM Aluno INNER INNER JOIN Historico ON Aluno.MatriculaAluno = Historico.MatriculaAluno
GROUP BY Aluno.MatriculaAluno;
```

Resultado:

```
MATRÍCULA MÉDIA_NOTAS
--------- -----------
1001      7.25
1002      7.5
1003      9
```

Elapsed: 00:00:00.002
3 rows selected.

	MATRÍCULA	MÉDIA_NOTAS
1	1001	7.25
2	1002	7.5
3	1003	9

4.6. Listar todas as disciplinas cadastradas e a quantidade de alunos matriculados em 20252.

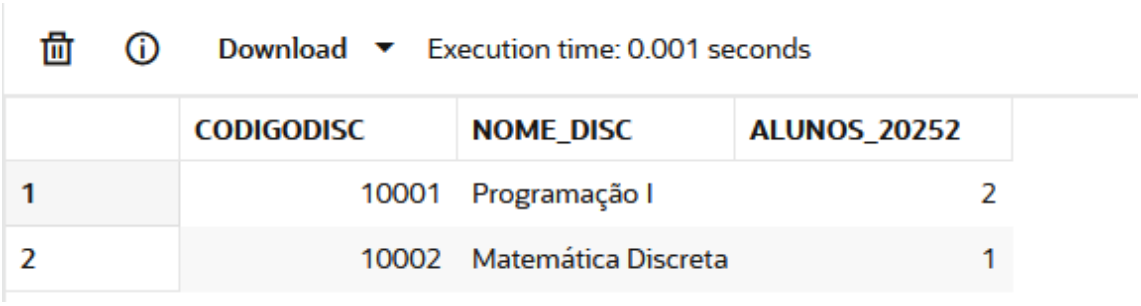
SQL utilizada:

```
SELECT Disciplina.CodigoDisc, Disciplina.Nome AS Nome_Disc, COUNT(*) AS Alunos_20252
FROM Disciplina INNER JOIN Historico ON Disciplina.CodigoDisc = Historico.CodigoDisc
        INNER JOIN Aluno ON Aluno.MatriculaAluno = Historico.MatriculaAluno
WHERE Aluno.Status = 'matriculado'
AND Historico.Semestre = 20252
GROUP BY Disciplina.CodigoDisc, Disciplina.Nome;
```

Resultado:

```
CODIGODISC NOME_DISC           ALUNOS_20252
---------- -------------------- ------------
10001      Programação I        2
10002      Matemática Discreta  1

Elapsed: 00:00:00.001
2 rows selected.
```



	CODIGODISC	NOME_DISC	ALUNOS_20252
1	10001	Programação I	2
2	10002	Matemática Discreta	1

4.7. Listar as disciplinas que não tiveram alunos matriculados em 20252.

SQL utilizada:

```
SELECT Disciplina.Nome AS Nome_Disc
FROM Disciplina
WHERE Disciplina.CodigoDisc NOT IN (
   SELECT Historico.CodigoDisc FROM Historico WHERE Historico.Semestre = 20252
)
GROUP BY Disciplina.CodigoDisc, Disciplina.Nome;
```

Resultado:

```
NOME_DISC
--------------------
Física Básica
Algoritmos
Banco de Dados II

Elapsed: 00:00:00.003
3 rows selected.
```

	NOME_DISC
1	Física Básica
2	Algoritmos
3	Banco de Dados II

4.8. Listar os professores que ministraram todas as disciplinas cadastradas no banco de dados.

SQL utilizada:

```
SELECT Professor.Nome
FROM Professor INNER JOIN Turma ON Professor.NumeroProf = Turma.NumeroProf
GROUP BY Professor.NumeroProf, Professor.Nome
HAVING COUNT(DISTINCT Turma.CodigoDisc) = (SELECT COUNT(DISTINCT Historico.CodigoDisc) FROM Historico);
```

Resultado:

0 rows selected.

NOME
No items to display.

4.9. Total de carga horária já cursada por cada aluno.

SQL utilizada:

```
SELECT Aluno.Nome AS Matricula, SUM(Disciplina.CargaHoraria) AS carga_horária_total
FROM Aluno INNER JOIN Historico ON Aluno.MatriculaAluno = Historico.MatriculaAluno
        INNER JOIN Disciplina ON Historico.CodigoDisc = Disciplina.CodigoDisc
GROUP BY Aluno.MatriculaAluno, Aluno.Nome;
```

Resultado:

```
MATRICULA       CARGA_HORÁRIA_TOTAL
--------------- -------------------
Joao Pereira    120
Maria Lima      60
Pedro Almeida   30

Elapsed: 00:00:00.021
3 rows selected.
```

	MATRICULA	CARGA_HORÁRIA_T
1	Joao Pereira	120
2	Maria Lima	60
3	Pedro Almeida	30

4.10. Semestres em que houve mais de 5 alunos matriculados em cada disciplina ofertada no semestre.

SQL utilizada:

```
SELECT DISTINCT semestre
FROM (
  SELECT t.semestre,
        t.codigoDisc,
        COUNT(DISTINCT h.matriculaAluno) AS qtd_alunos
  FROM TURMA t
  JOIN HISTORICO h ON t.codigoDisc = h.codigoDisc AND t.semestre = h.semestre
  GROUP BY t.semestre, t.codigoDisc
  HAVING COUNT(DISTINCT h.matriculaAluno) > 5
);
```

Resultado:

0 rows selected.

SEMESTRE
No items to display.