

Banco de Dados Relacional

PROFº DRº FRANCISCO DOUGLAS. L ABREU

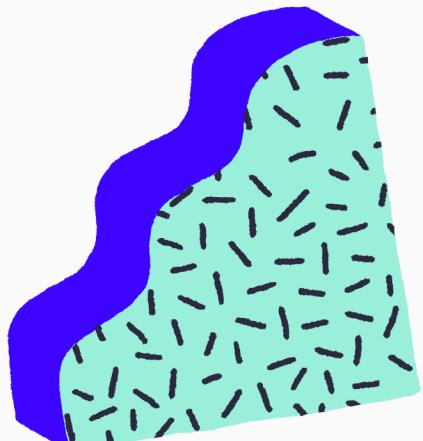
MSC E PHD EM ENGENHARIA BIOMÉDICA
ESP. EM ANÁLISE DE DADOS

FRANCISCO.ABREU@FATEC.SP.GOV.BR



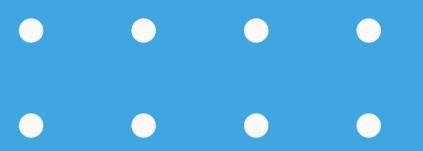
• • • •

• • • •





REVISÃO DA ATIVIDADE PONTUADA



Atividade

1. Criar uma Tabela em DDL para armazenar os dados de **Alunos**



- Não será permitido dois alunos terem o mesmo RA;
- O RA é o código que será referenciado por outras tabelas;
- Os campos nome, telefone, email e data nascimento são palavras;
- O campo nome e email são obrigatórios para todos alunos;

Atividade

2. Criar uma Tabela em DDL para armazenar os dados de **Disciplinas***
3. Criar uma Tabela em DDL para armazenar as **Notas** dos Alunos*
4. Fazer o Modelo de Entidade-Relacionamento (MER) e Modelo Relacional Lógico

***Os campos ficarão a sua definição, não esqueça das chaves primárias e estrangeiras**

****Pode ser desenvolvido no Visio, BrModelo, Bizagi, PowerPoint - Entregar apenas a Figura em JPG ou PNG**

Nota: 0 a 10 (compondo 40% da N1)

A Não entrega da atividade pelo o Aluno no prazo estimulado será pontuado 0 pts

Atividade



```
CREATE TABLE `fatec`.`aluno` (
  `id_ra` INT NOT NULL,
  `nome` VARCHAR(255),
  `dt_nasc` DATETIME,
  `email` VARCHAR(255) NOT NULL,
  `telefone` VARCHAR(14),
  PRIMARY KEY (`id_ra`)

);
```

Atividade

2. Criar uma Tabela em DDL para armazenar os dados de **Disciplinas***

```
● ● ●  
CREATE TABLE `fatec`.`disciplinas` (  
    `id_Disciplinas` INT NOT NULL,  
    `nome` VARCHAR(255) NOT NULL,  
    `sigla` VARCHAR(25) ,  
    `ementa` LONGTEXT NOT NULL,  
    `objetivo` MEDIUMTEXT NOT NULL,  
    `metodologia` LONGTEXT NOT NULL,  
    `bibliografia` LONGTEXT NOT NULL,  
    PRIMARY KEY (`id_Disciplinas`)  
);
```

Atividade

3. Criar uma Tabela em DDL para armazenar as Notas dos Alunos*



```
CREATE TABLE `fatec`.`notas` (
  `N1` FLOAT NOT NULL,
  `N2` FLOAT NOT NULL,
  `N3` FLOAT NOT NULL,
  `MF` FLOAT NOT NULL,
  `id_Disciplinas` INT NOT NULL,
  `id_Aluno` INT NOT NULL,
  FOREIGN KEY (`id_Disciplinas`) REFERENCES `fatec`.`disciplinas`(id_Disciplinas),
  FOREIGN KEY (`id_Aluno`) REFERENCES `fatec`.`aluno`(id_ra)
);
```

Atividade

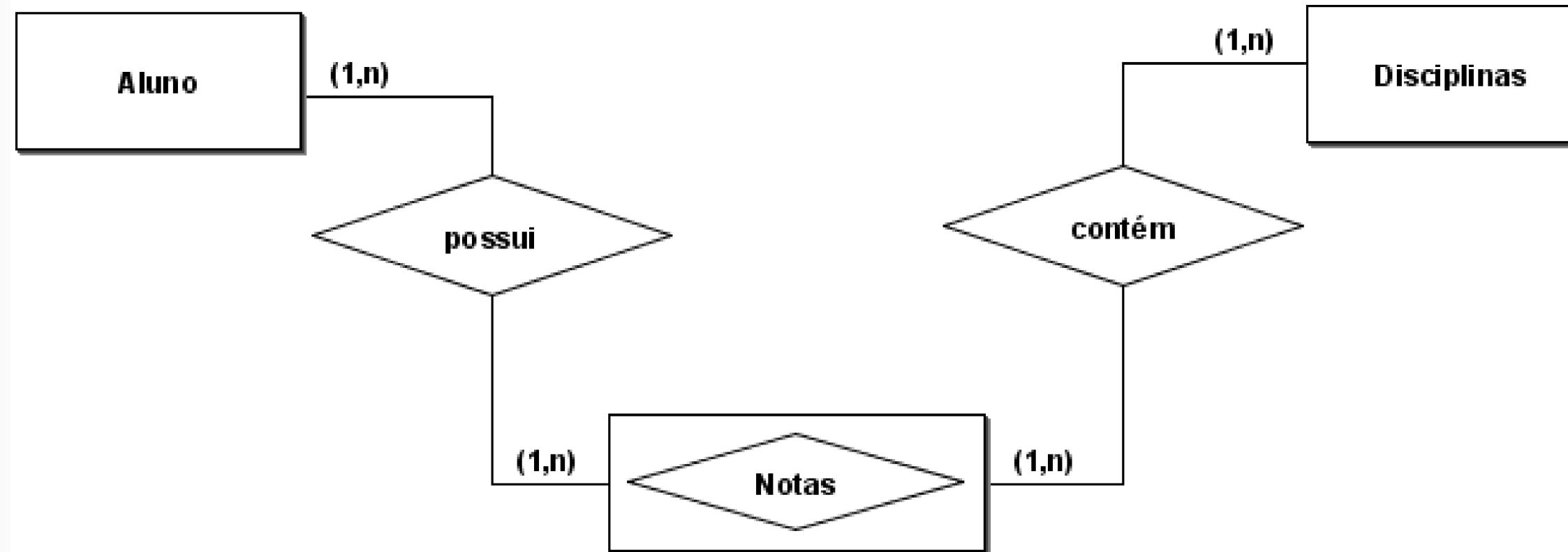
3. Fazer o Modelo de Entidade-Relacionamento (MER) e Modelo Relacional Lógico *

Possíveis Diagramas Corretos:

✓ 1



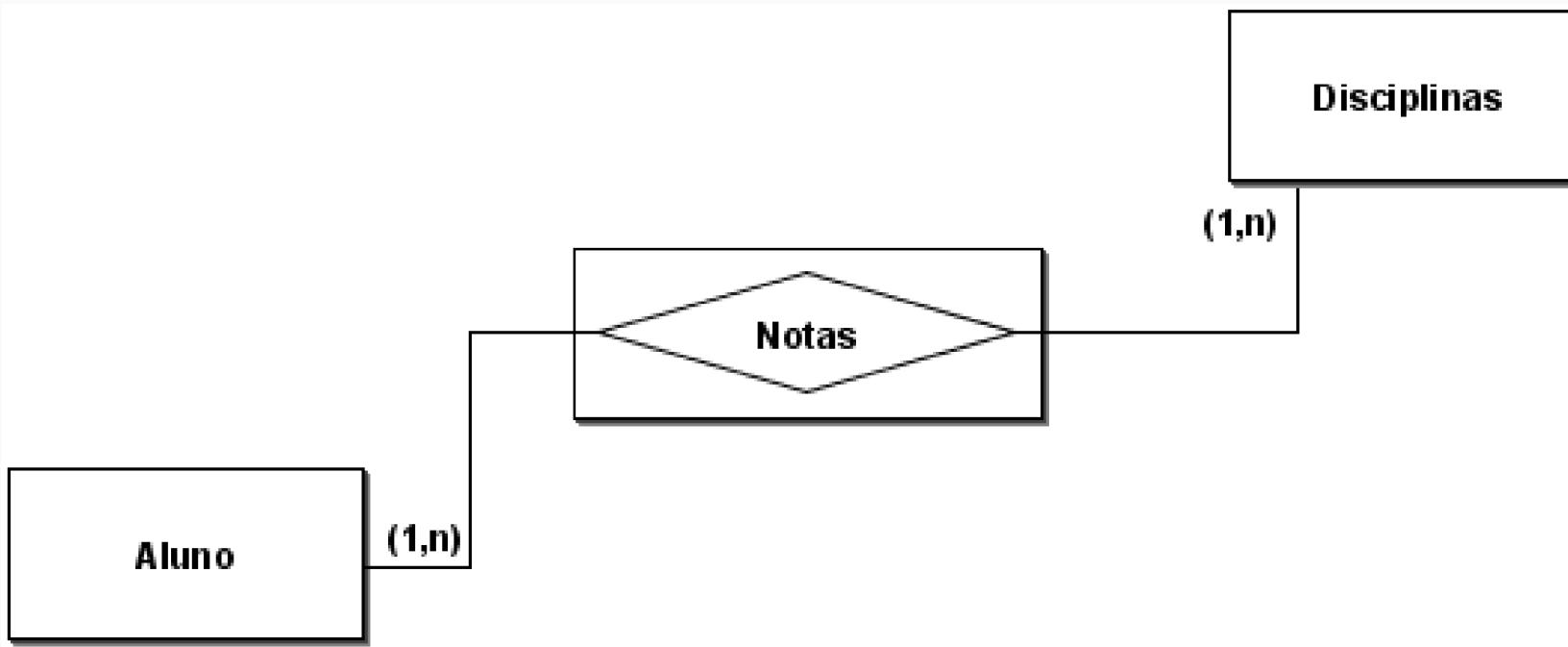
✓ 2



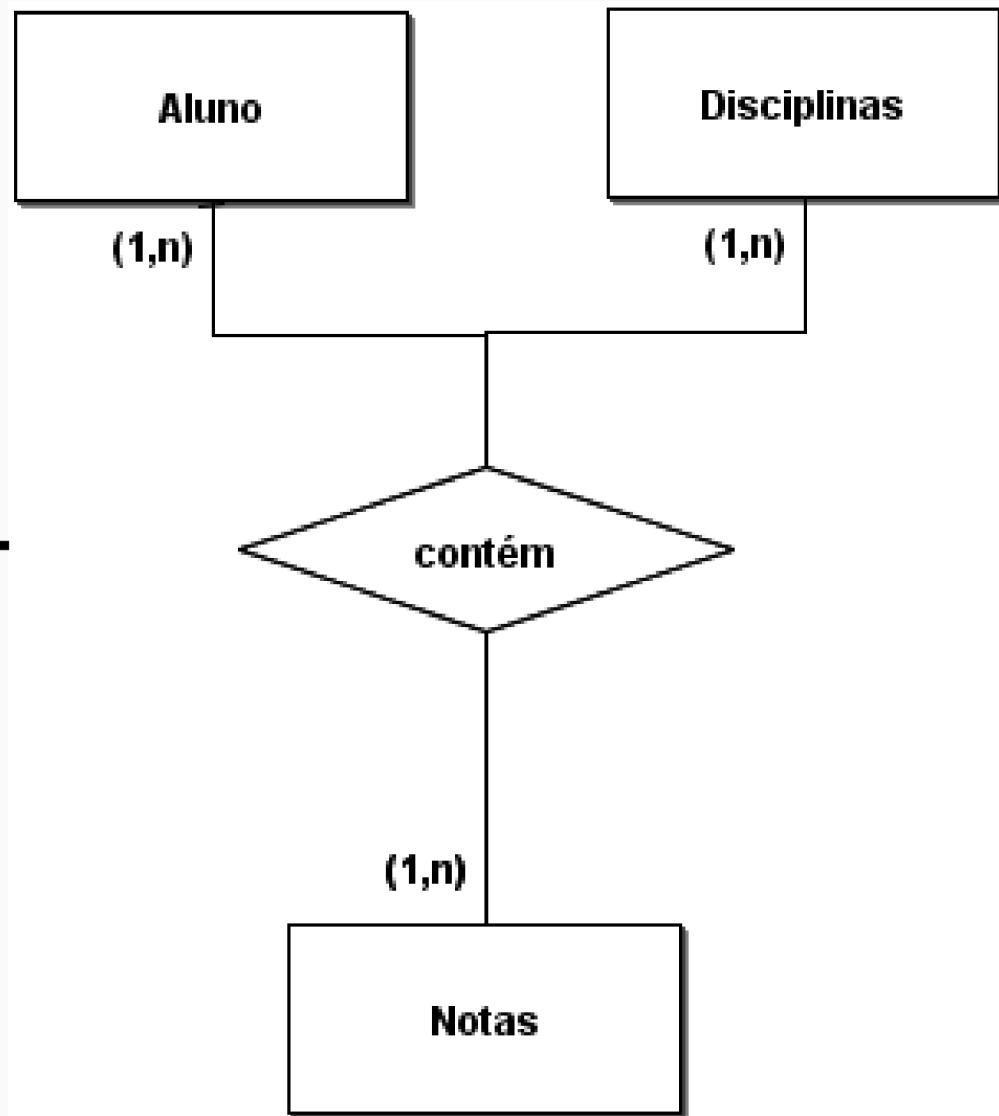
Atividade

✓

3

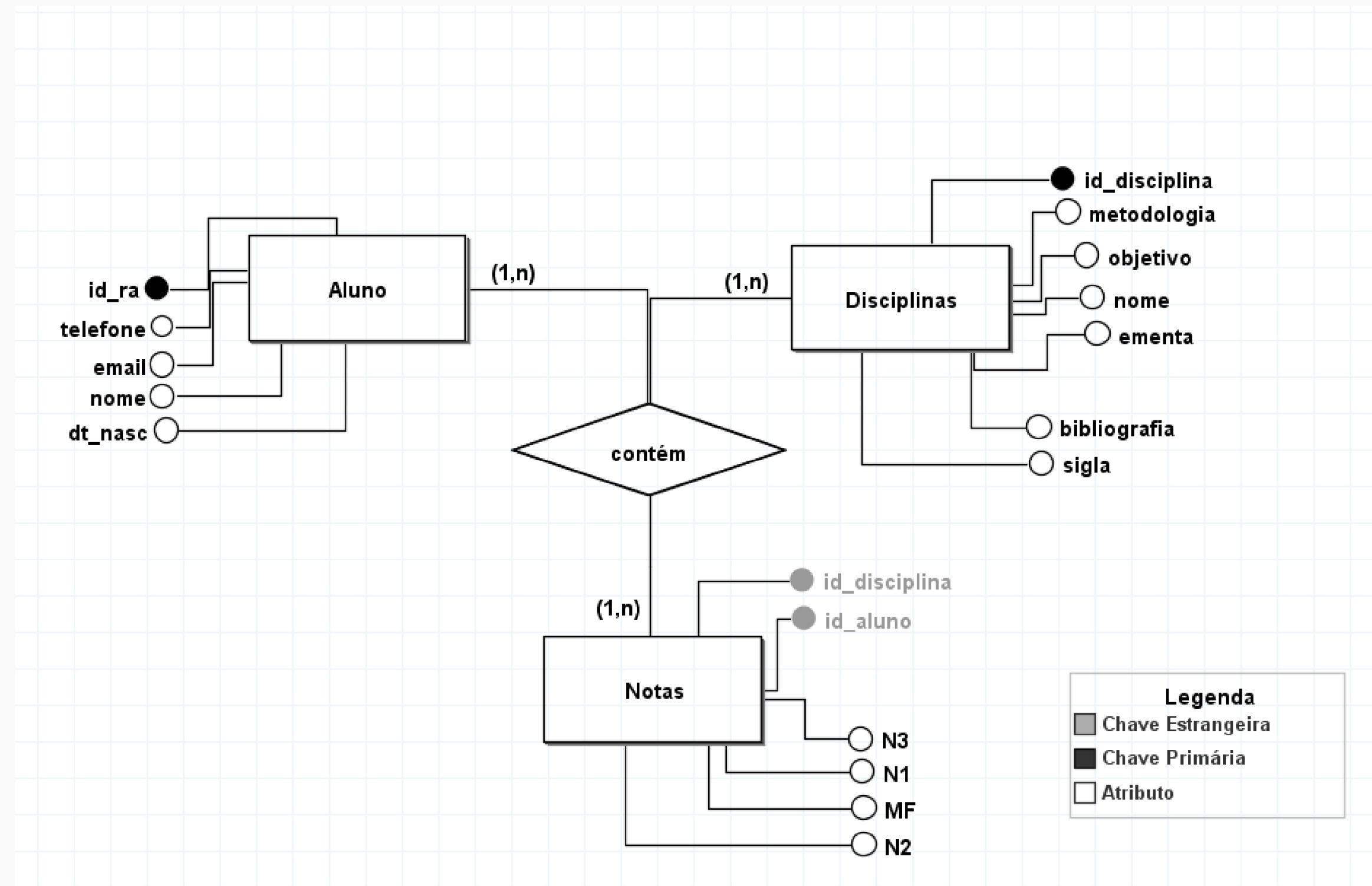


✓ 4



Atividade

✓ 5

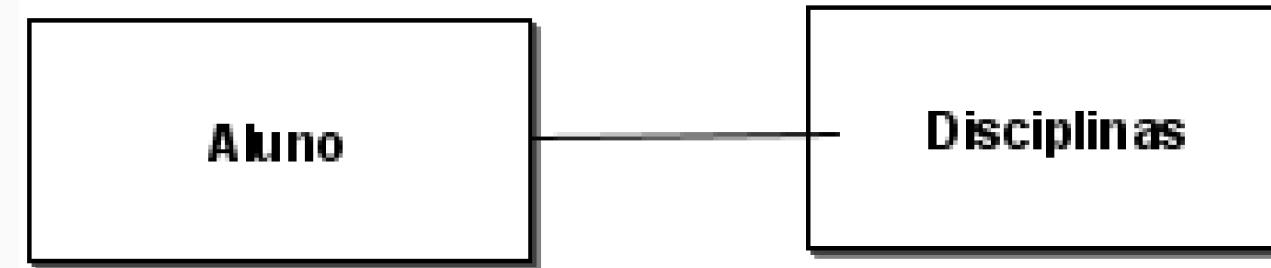


Atividade

X

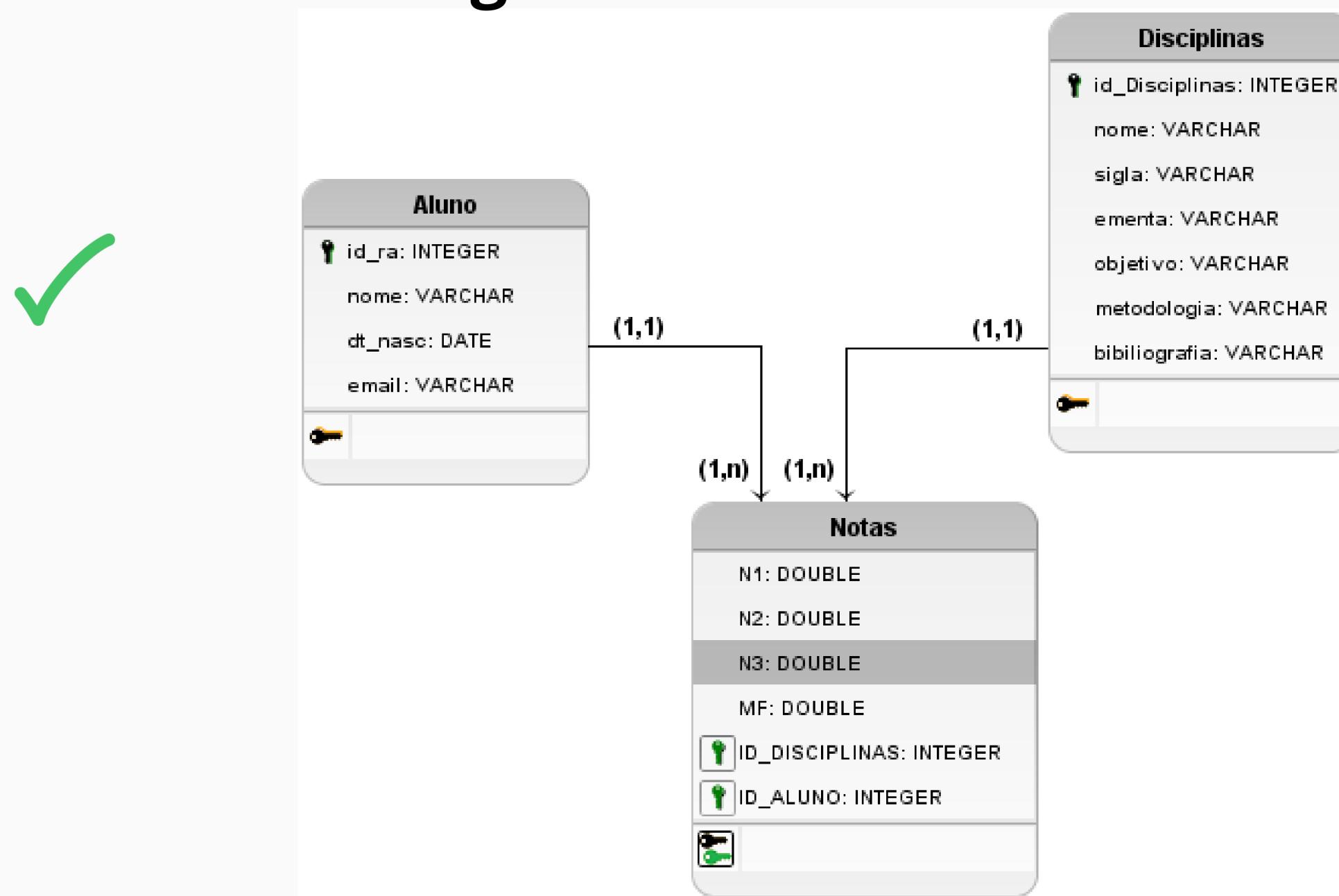


X



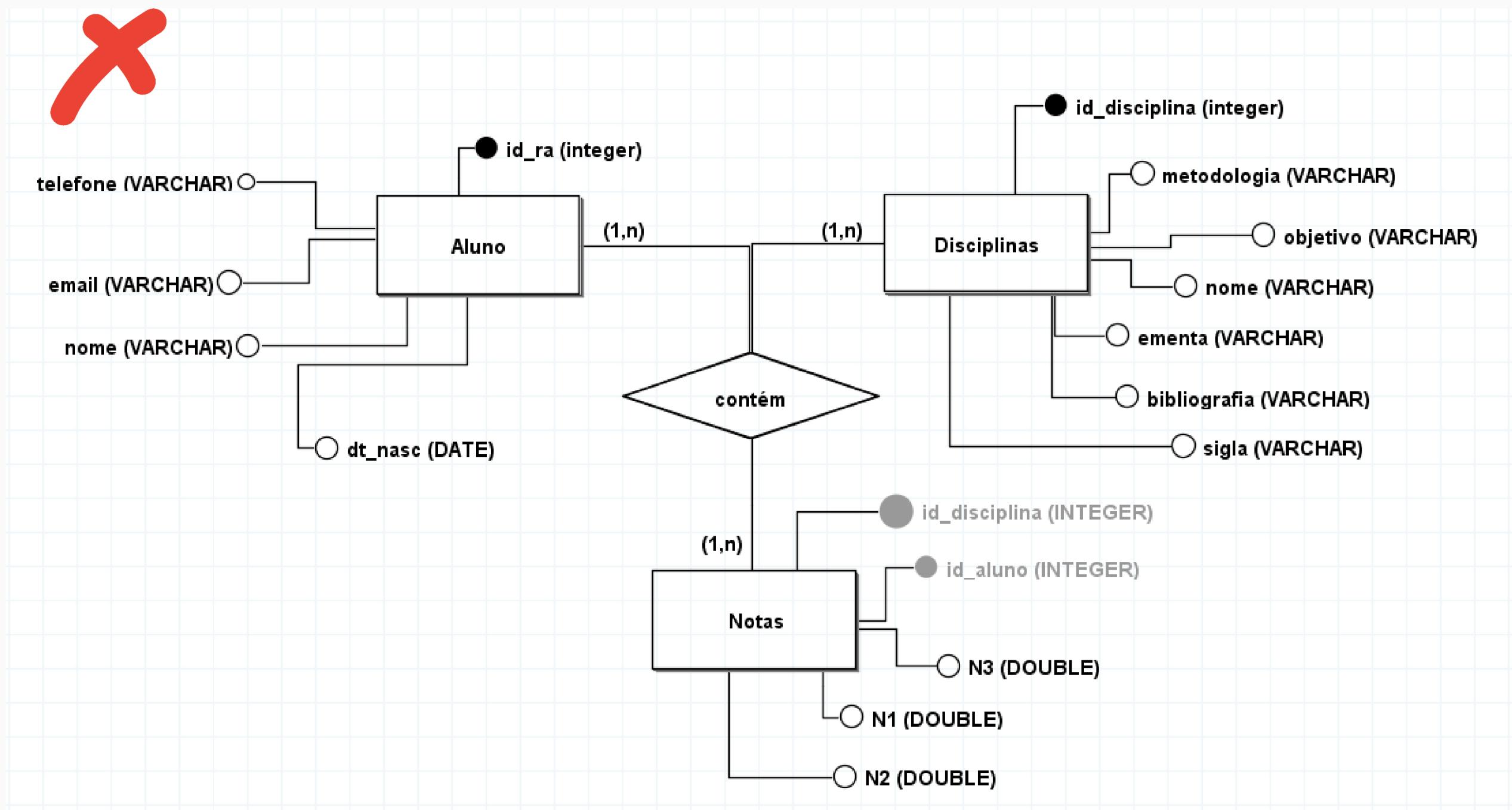
Atividade

Modelo Relacional Lógico Correto:



Atividade

Modelo Relacional Lógico incorreto:



Alguma Dúvida?



DDL

**CREATE
ALTER
DROP
COMMENT
SHOW
DESCRIBE**

DML

DQL

DTL

DTL

TIPOS DE ATRIBUTO

TIPOS DE DADOS: NUMÉRICOS

TIPO	DESCRIÇÃO
BIT	o mesmo que TINYINT
BOOL	o mesmo que TINYINT
SMALLINT[(M)]	inteiros pequenos
INT[(M)]	inteiros regulares
INTEGER[(M)]	o mesmo que INT
BIGINT[(M)]	inteiros grandes
FLOAT(precisão)	números de ponto flutuante de precisão simples ou dupla
FLOAT[(M,D)]	números de ponto flutuante de precisão simples. O mesmo que FLOAT(4)
DOUBLE[(M,D)]	números de ponto flutuante de precisão dupla. O mesmo que FLOAT(8)
DOUBLE	O mesmo que DOUBLE[(M,D)]
PRECISION[(M,D)]	O mesmo que DOUBLE[(M,D)]
REAL[(M,D)]	O mesmo que DOUBLE[(M,D)]
DECIMAL[(M,D)]	número de ponto flutuante armazenado como char
NUMERIC[(M,D)]	O mesmo que DECIMAL
DEC[(M,D)]	O mesmo que DECIMAL

TIPO DE DADOS: DATA E HORA

TIPO	DESCRÍÇÃO
DATE	data. Exibido como YYYY-MM-DD
TIME	hora. Exibido como HH:MM:SS
DATETIME	data e hora. Exibido como YYYY-MM-DD HH:MM:SS
TIMESTAMP[(M)]	registro de data e hora útil para transações. Os formatos de exibição podem ser: TIMESTAMP YYYYMMDDHHMMSS TIMESTAMP(14) YYYYMMDDHHMMSS TIMESTAMP(12) YYMMDDHHMMSS TIMESTAMP(10) YYMMDDHHMM TIMESTAMP(8) YYYYMMDD TIMESTAMP(6) YYMMDD TIMESTAMP(4) YYMM TIMESTAMP(2) YY
YEAR[(2)]	ano
YEAR[(4)]	ano

TIPO DE DADOS: TEXTO OU LITERAL

TIPO	DESCRIÇÃO
[NATIONAL] CHAR(M) [BINARY]	string de comprimento fixo M . NATIONAL especifica que o conjunto de caracteres padrão (ANSI SQL) será utilizado. BINARY especifica que os dados devem ser tratados de modo a não haver distinção entre maiúsculas e minúsculas (o padrão é distinguir).
CHAR	o mesmo que CHAR(1)
[NATIONAL]	string de comprimento variável
VARCHAR(M) [BINARY]	string de tamanho variável. O mesmo que [BINARY].
TINYBLOB	BLOB pequeno
TINYTEXT	TEXT pequeno
BLOB	BLOB normal
TEXT	TEXT normal
MEDIUMBLOB	BLOB médio
MEDIUMTEXT	TEXT médio
LONGBLOB	BLOB longo
LONGTEXT	TEXT longo
ENUM('valor1','valor2',...)	armazenam um dos valores listados ou NULL

COMENTÁRIOS NO SQL

Comentários no SQL

Existem duas formas de se fazer um comentário na linguagem SQL:

ÚNICA LINHA

-- EXEMPLO 01 DE COMENTÁRIO EM UMA LINHA

#EXEMPLO 02 DE COMENTÁRIO EM UMA LINHA

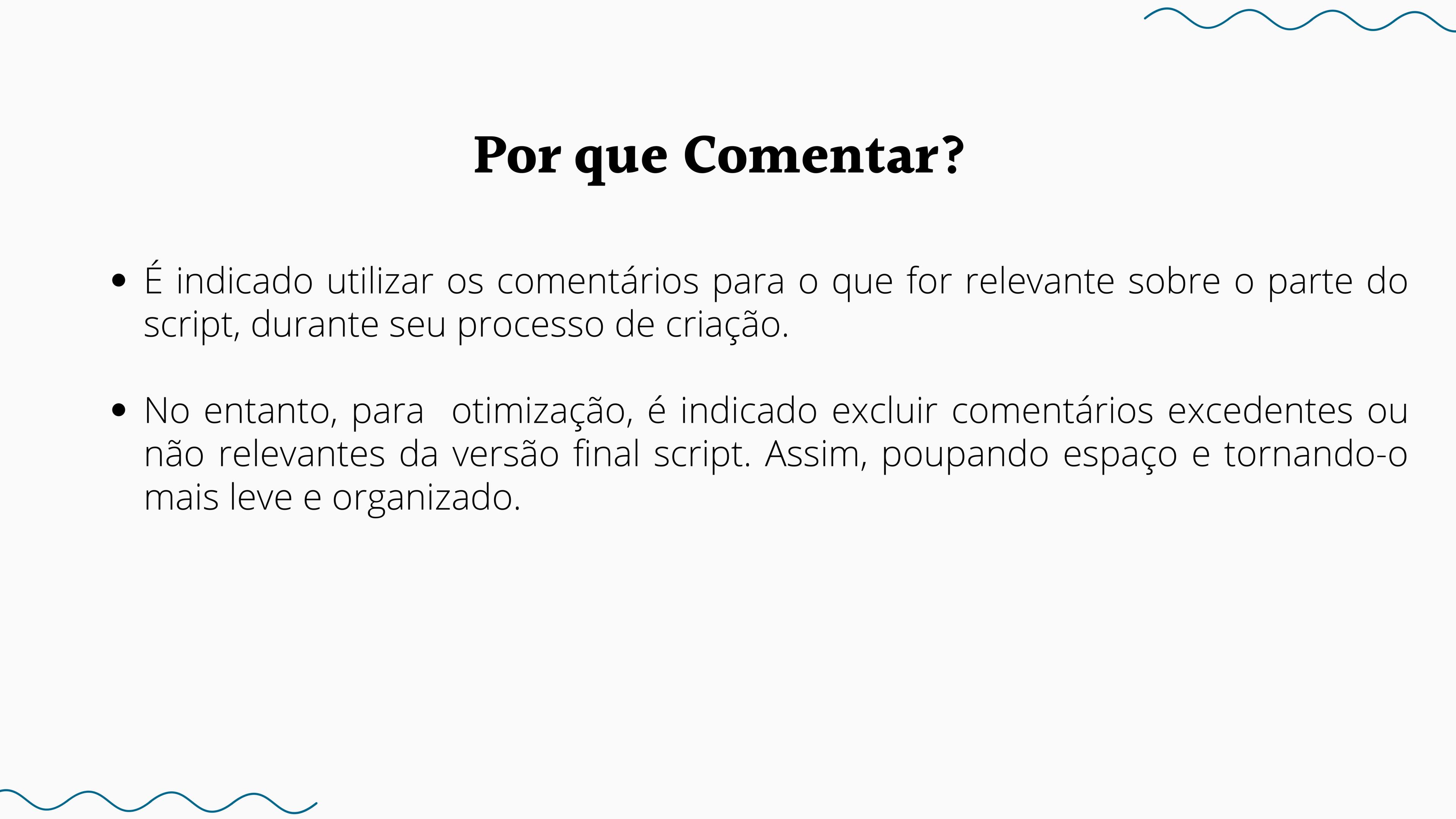
BLOCO DE LINHAS

```
/* EXEMPLO  
DE COMENTÁRIO EM  
MULTIPLAS  
LINHAS */
```



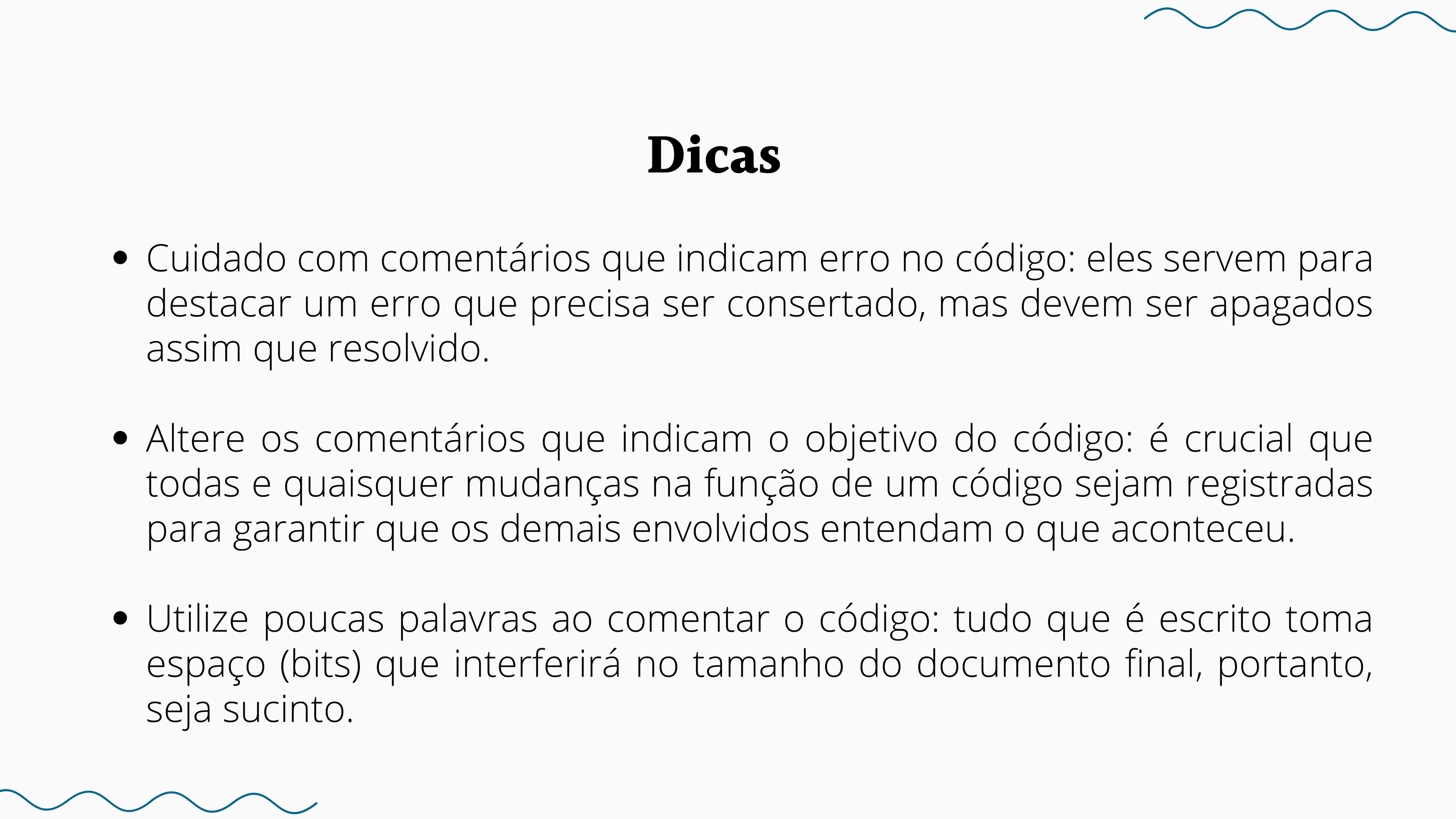
O que é Comentar?

- São informações, dicas, avisos, conselhos, que só podem ser vistos dentro do código-fonte, no caso o SQL.
 - Um comentário SQL é um espaço que o desenvolvedor tem para comunicar o que o script escrito está fazendo, como ele funciona e porque foi feito daquela forma. Logo, é possível fazer ajustes, aperfeiçoamentos e mudanças futuras, de maneira mais rápida.
- 



Por que Comentar?

- É indicado utilizar os comentários para o que for relevante sobre o parte do script, durante seu processo de criação.
- No entanto, para otimização, é indicado excluir comentários excedentes ou não relevantes da versão final script. Assim, poupando espaço e tornando-o mais leve e organizado.



Dicas

- Cuidado com comentários que indicam erro no código: eles servem para destacar um erro que precisa ser consertado, mas devem ser apagados assim que resolvido.
- Altere os comentários que indicam o objetivo do código: é crucial que todas e quaisquer mudanças na função de um código sejam registradas para garantir que os demais envolvidos entendam o que aconteceu.
- Utilize poucas palavras ao comentar o código: tudo que é escrito toma espaço (bits) que interferirá no tamanho do documento final, portanto, seja sucinto.



"CADÊ OS COMENTÁRIOS?"



"ACHEI QUE NÃO
PRECISAVA!"

imagegenerator.net

DATA DESCRIPTION LANGUAGE - DDL



Alterar Tabelas (ALTER TABLE)

O comando ALTER TABLE altera a definição de uma tabela existente. É necessário ser o dono da tabela para executar o comando ALTER TABLE, exceto para o super usuário. Existem várias formas alternativas:

ADD COLUMN

Esta forma adiciona uma nova coluna à tabela usando a mesma sintaxe do comando CREATE TABLE

SINTAXE

```
ALTER TABLE <NOME_DO_BANCO>.<NOME_DA_TABELA>
    ADD COLUMN <NOME_DO_CAMPO> <TIPO_E_TAMANHO_DE_DADOS> <ACEITAÇÃO>;
```





Parâmetros:

<NOME_DA_TABELA>

O nome da tabela a ser alterada

<NOME_DO_CAMPO>

O nome da coluna/campo a ser alterada/criada na nova tabela

<TIPO_E_TAMANHO_DE_DADOS>

O tipo de dado da coluna e o tamanho máximo que pode assumir

<ACEITAÇÃO>

Tipo de aceitação do dado, ou seja, se ele pode ou não receber valores nulos ou não.

NOT NULL - Valores nulos não são permitidos na coluna. NULL - Valores nulos são permitidos na coluna. Este é o padrão



DROP COLLUM

Esta forma exclui uma coluna da tabela. Note que os índices e as restrições da tabela que referenciam a coluna também serão automaticamente excluídos. Será necessário especificar CASCADE se algum objeto fora da tabela depender da coluna, por exemplo, chaves estrangeiras, visões, etc.

SINTAXE

```
ALTER TABLE <NOME_DO_BANCO>.<NOME_DA_TABELA>
DROP COLLUM <NOME_DO_CAMPO> [RESTRICT | CASCADE] ;
```

Parâmetros:

<CASCADE>

Exclui, automaticamente, os objetos dependentes da **Coluna** ou restrição excluída (por exemplo, visões fazendo referência à coluna)

<RESTRICT>

Recusa excluir a **Coluna** ou a restrição se existirem objetos dependentes. Este é o comportamento padrão.

RENAME

Esta forma altera o nome de uma tabela.

SINTAXE

```
ALTER TABLE <NOME_DO_BANCO>.<NOME_DA_TABELA>
    RENAME TO <NOVO_NOME_DA_TABELA>;
```

CHANGE COLUMN

Esta forma altera nome de uma coluna da tabela.

SINTAXE

```
ALTER TABLE <NOME_DO_BANCO>.<NOME_DA_TABELA>
    CHANGE COLUMN <NOME_DO_CAMPO_ANTERIOR><NOME_DO_CAMPO_NOVO>;
```

MODIFY

Esta forma altera/modifica o tipo de dado ou aceitação em uma coluna da tabela.

Estas formas mudam a definição da coluna para permitir valores nulos ou para rejeitar valores nulos. A forma NOT NULL somente pode ser usada quando não existem valores nulos na coluna

SINTAXE

```
ALTER TABLE <NOME_DO_BANCO>.<NOME_DA_TABELA>
    MODIFY <NOME_DO_CAMPO><TIPO_E_TAMANHO_DE_DADOS>;
```

EXEMPLOS

1. ADICIONAR UMA COLUNA DO TIPO VARCHAR EM UMA TABELA:

ALTER TABLE ALUNO

ADD COLUMN GENERO VARCHAR(10);

2. EXCLUIR UMA COLUNA DE UMA TABELA:

ALTER TABLE ALUNO

DROP COLUMN EMAIL RESTRICT;

3. PARA MUDAR O NOME DE UMA COLUNA EXISTENTE:

ALTER TABLE ALUNO

CHANGE COLUMN NOME NOME_COMPLETO VARCHAR(255);

4. ADICIONAR UMA RESTRIÇÃO NOT NULL A UMA COLUNA:

ALTER TABLE ALUNO

MODIFY NOME_COMPLETO VARCHAR (200) NOT NULL

5. MUDAR O NOME DE UMA TABELA EXISTENTE:

ALTER TABLE ALUNO

RENAME TO ESTUDANTES;

Remoção de Tabelas (DROP TABLE)

O comando DROP TABLE remove tabelas do banco de dados. Somente o criador pode remover a tabela. Isto é, remove todos os índices, regras, gatilhos e restrições existentes na tabela.

SINTAXE

```
DROP TABLE <NOME_DO_BANCO>.<NOME_DA_TABELA> [CASCADE | RESTRIC];
```

Parâmetros:

<CASCADE>

Exclui, automaticamente, os objetos dependentes da **Tabela** ou restrição excluída (por exemplo, visões fazendo referência à coluna)

<RESTRICT>

Recusa excluir a **Tabela** ou a restrição se existirem objetos dependentes. Este é o comportamento padrão.

EXEMPLOS

1. REMOVER UMA TABELA NO BANCO DE DADOS:

DROP TABLE ALUNO;

2. REMOVER UM CONJUNTO DE TABELAS (TABELAS SEM RELACIONAMENTO)

DROP TABLE ALUNO, DISCIPLINAS;

3. REMOVER UM CONJUNTO DE TABELAS RELACIONADAS É NECESSÁRIO CITAR
TODAS AS TABELAS

DROP TABLE NOTAS, ALUNO, DISCIPLINAS;

Atenção! Remover primeiro as tabelas fracas (possui chave estrangeira) e logo depois as tabelas forte

Alguma Dúvida?

Atividade

- Criar o script de criação das tabelas (DDL) para o diagrama de entidade-relacionamento
- Criar o script de exclusão de tabelas (DDL) para o diagrama de entidade-relacionamento
- Alterar a coluna nome da Tabela Cliente para Nome_Completo
- Adicionar uma coluna na Tabela Veículo, denominado Ano

Entregar o Script no formato **.SQL**

Prazo: às 23h59

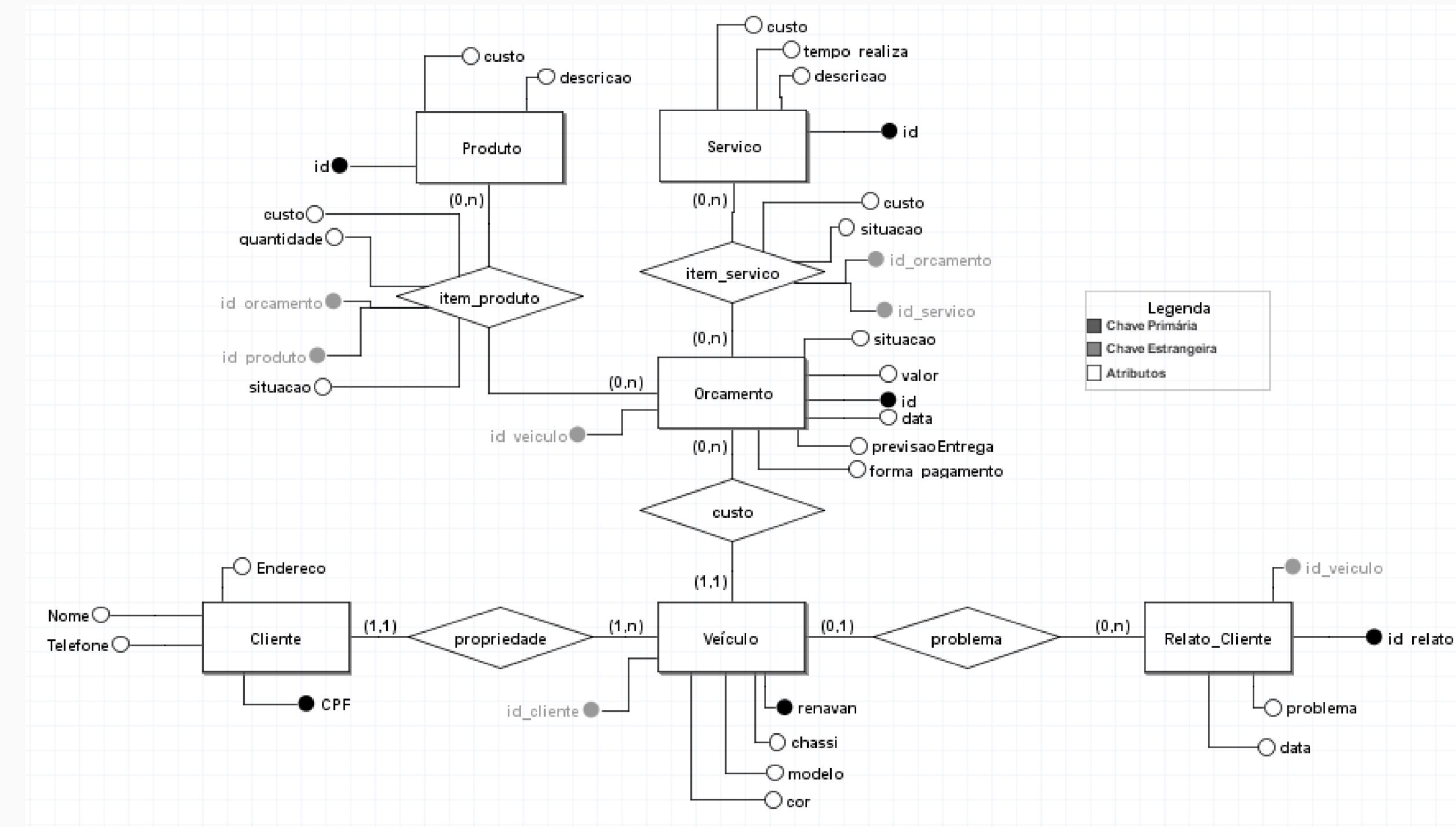
Nota: 0 a 10 (compondo a nota de Atividades da N1)

A Não entrega da atividade pelo o Aluno no prazo estimulado será pontuado 0 pts

Atividade Liberada No Teams

Atividade

Oficina
Mecânica





Copyright © 2023 Profº Drº Francisco Douglas Lima Abreu

Todos direitos reservados. Reprodução ou divulgação total ou parcial deste documento é expressamente proibido sem o consentimento formal, por escrito ao autor