

Banco de Dados Ciclos de Estudos

Ciclos de Estudo

—

Por Antonio Welton Ribeiro Souza

Para que se analise a eficiência do aprendizado é fundamental manter informações sobre o tempo despendido nos estudos.

O que se registra no banco de dados Ciclo de Estudo?

Ciclo de Estudo é um banco de dados simples com a finalidade principal de permitir ao estudante registrar o horário que começou e que parou estudar, além disso permite analisar o assunto estudado bem como a disciplina, curso, área do conhecimento e o professor. Todos esses critérios serão associados permitindo uma visão ampla de cada ciclo de estudo.

Ferramentas Utilizadas

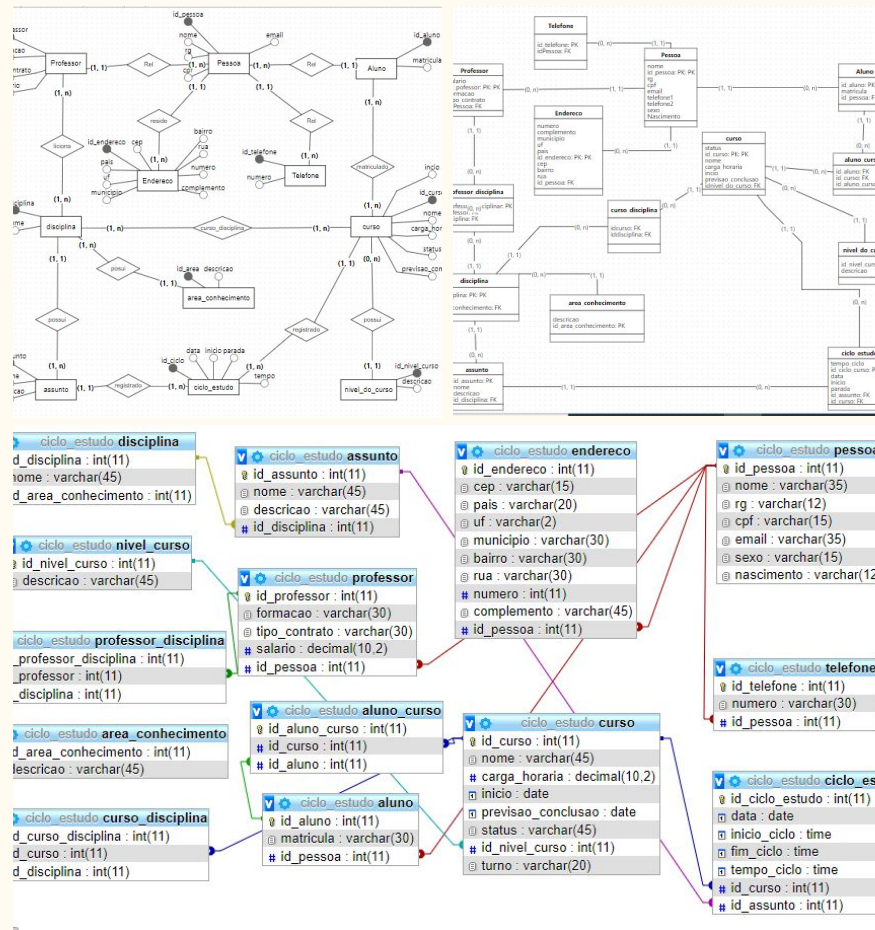
Todas são gratuitas!

- br modelo-web
- XAMPP
- MYSQL
- MYSQL WorkBench



Fases:

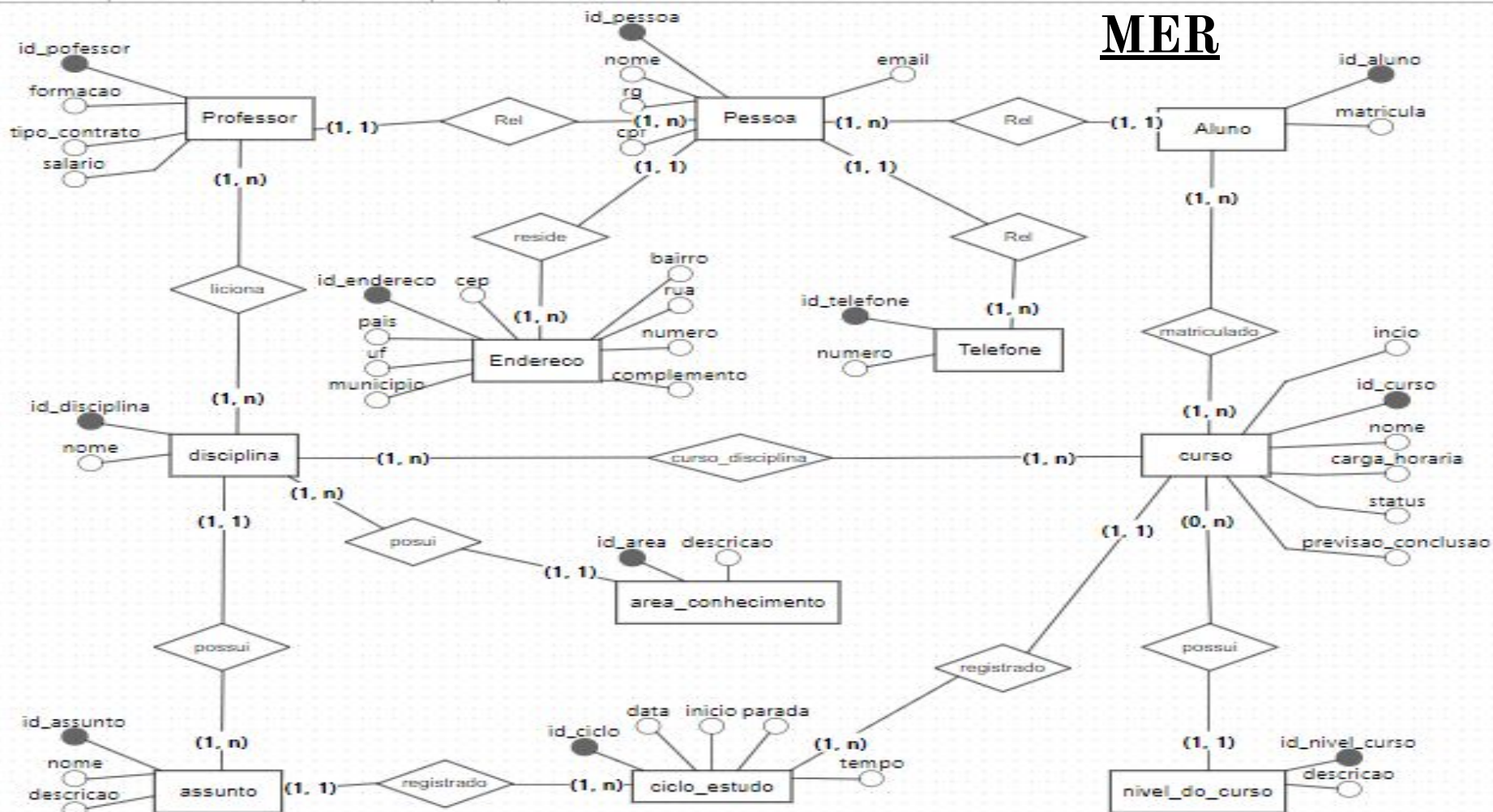
1. Modelagem Conceitual
2. Modelagem Lógica
3. Modelagem Física



Modelagem Conceitual

Na modelagem conceitual observa-se o Modelo Entidade Relacionamento(MER). Cria-se um diagrama com a finalidade de demonstrar as entidades, os atributos e os graus de relacionamento(1 para 1, 1,1; 1 para muitos, 1,m; muitos para muitos, m,m).

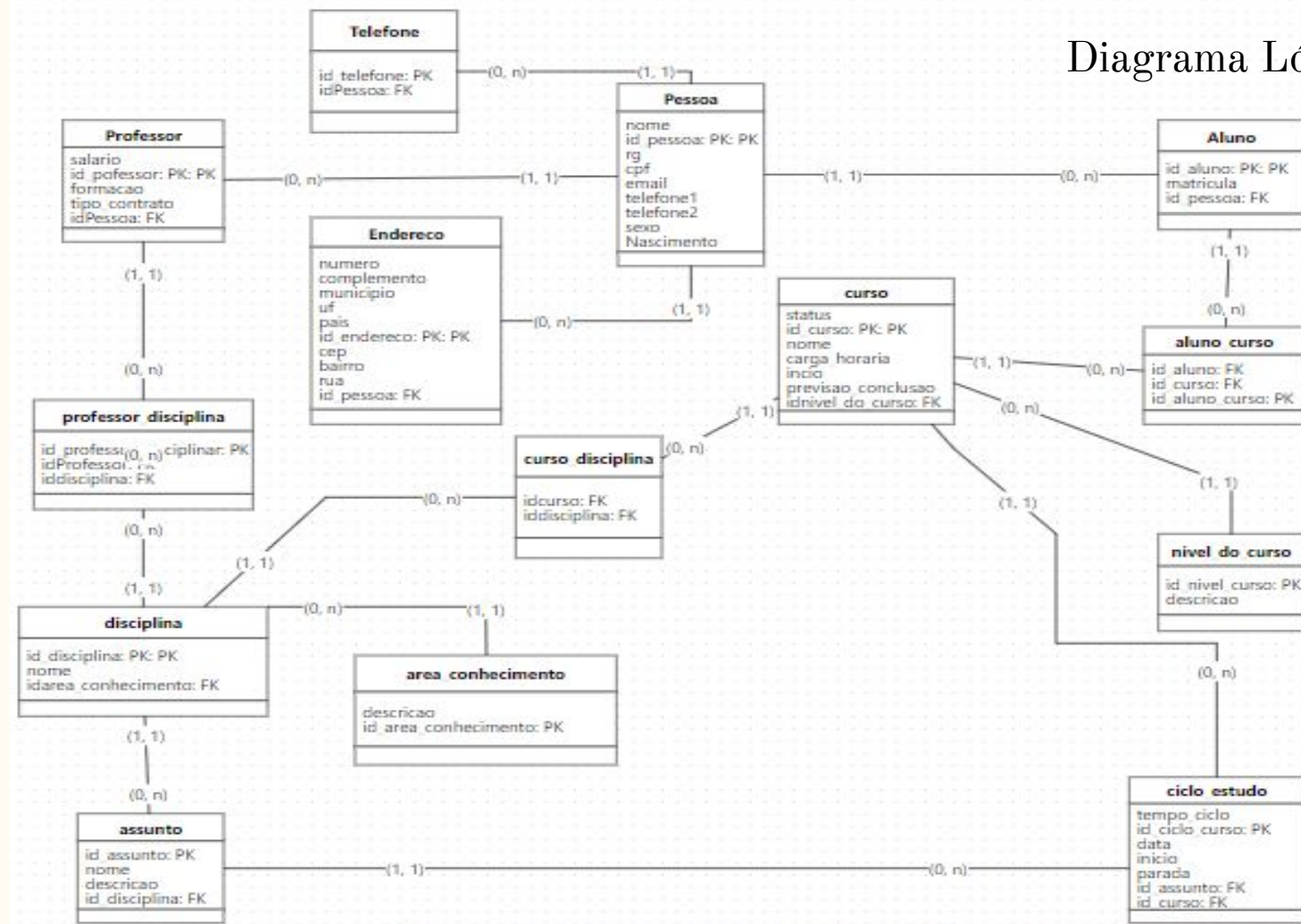
MER



Modelagem Lógica.

Agora se traduz as entidades, seus atributos e relacionamentos em tabelas compostas por linhas e colunas com a indicação de seus atributos chaves. Pode-se representar essa etapa por meio do diagrama lógico:

Diagrama Lógico



Modelagem Física.

Etapas de criação das tabelas em um SGBD, usando a linguagem Structured Query Language(SQL).



```
CREATE DATABASE ciclo_estudo;

use ciclo_estudo;

CREATE TABLE `area_conhecimento` (
  `id_area_conhecimento` int NOT NULL AUTO_INCREMENT,
  `descricao` varchar(45) NOT NULL,
  PRIMARY KEY (`id_area_conhecimento`)
);

CREATE TABLE `nivel_curso` (
  `id_nivel_curso` int NOT NULL AUTO_INCREMENT,
  `descricao` varchar(45) NOT NULL,
  PRIMARY KEY (`id_nivel_curso`)
);

CREATE TABLE `disciplina` (
  `id_disciplina` int NOT NULL AUTO_INCREMENT,
  `nome` varchar(45) NOT NULL,
  `id_area_conhecimento` int DEFAULT NULL,
  PRIMARY KEY (`id_disciplina`),
  KEY `id_area_conhecimento` (`id_area_conhecimento`),
  CONSTRAINT `disciplina_ibfk_1` FOREIGN KEY (`id_area_conhecimento`) REFERENCES `area_conhecimento` (`id_area_conhecimento`)
);
```

```
CREATE TABLE `curso` (  
  `id_curso` int NOT NULL AUTO_INCREMENT,  
  `nome` varchar(45) NOT NULL,  
  `carga_horaria` decimal(10,2) DEFAULT NULL,  
  `inicio` date DEFAULT NULL,  
  `previsao_conclusao` date DEFAULT NULL,  
  `status` varchar(45) DEFAULT NULL,  
  `id_nivel_curso` int DEFAULT NULL,  
  `id_area_conhecimento` int DEFAULT NULL,  
  PRIMARY KEY (`id_curso`),  
  KEY `id_nivel_curso` (`id_nivel_curso`),  
  CONSTRAINT `curso_ibfk_1` FOREIGN KEY (`id_nivel_curso`) REFERENCES `nivel_curso` (`id_nivel_curso`),  
  KEY `id_area_conhecimento` (`id_area_conhecimento`),  
  CONSTRAINT `curso_ibfk_2` FOREIGN KEY (`id_area_conhecimento`) REFERENCES `area_conhecimento` (`id_area_conhecimento`)  
);  
  
CREATE TABLE `assunto` (  
  `id_assunto` int NOT NULL AUTO_INCREMENT,  
  `nome` varchar(45) NOT NULL,  
  `descricao` varchar(45) DEFAULT NULL,  
  `id_disciplina` int NOT NULL,  
  PRIMARY KEY (`id_assunto`),  
  KEY `id_disciplina` (`id_disciplina`),  
  CONSTRAINT `assunto_ibfk_1` FOREIGN KEY (`id_disciplina`) REFERENCES `disciplina` (`id_disciplina`)  
);
```

```
create table pessoa(  
    id_pessoa int primary key auto_increment,  
    nome varchar(35) not null,  
    rg varchar(12),  
    cpf varchar(15) not null,  
    email varchar(35),  
    sexo varchar(15),  
    nascimento varchar(12) not null  
);  
  
create table endereco(  
    id_endereco int primary key auto_increment,  
    cep varchar(15) not null,  
    pais varchar(20) not null,  
    uf varchar(2) not null,  
    municipio varchar(30) not null,  
    bairro varchar(30) not null,  
    rua varchar(30) not null,  
    numero int not null,  
    complemento varchar(45),  
    id_pessoa int not null,  
    foreign key(id_pessoa) references pessoa(id_pessoa));
```

```
create table telefone(  
    id_telefone int primary key auto_increment,  
    numero varchar(30) not null,  
    id_pessoa int not null,  
    foreign key(id_pessoa) references pessoa(id_pessoa));  
create table professor(  
    id_professor int primary key auto_increment,  
    formacao varchar(30) not null,  
    tipo_contrato varchar(30) not null,  
    salario decimal(10,2),  
    id_pessoa int not null,  
    foreign key (id_pessoa) references pessoa(id_pessoa)  
);  
create table aluno(  
    id_aluno int primary key auto_increment,  
    matricula varchar(30) not null,  
    id_pessoa int not null,  
    foreign key(id_pessoa) references pessoa(id_pessoa)  
);
```



```
-- Tabelas associativas
create table aluno_curso(
    id_aluno_curso int primary key auto_increment,
    id_curso int not null,
    id_aluno int not null,
    foreign key(id_aluno) references aluno(id_aluno),
    foreign key(id_curso) references curso(id_curso)
);

create table professor_disciplina(
    id_professor_disciplina int primary key auto_increment,
    id_professor int not null,
    id_disciplina int not null,
    foreign key(id_professor) references professor(id_professor),
    foreign key(id_disciplina) references disciplina(id_disciplina)
);
```

```
-- Tabelas associativas
create table curso_disciplina(
    id_curso_disciplina int primary key auto_increment,
    id_curso int not null,
    id_disciplina int not null,
    foreign key(id_curso) references curso(id_curso),
    foreign key(id_disciplina) references disciplina(id_disciplina));

CREATE TABLE `ciclo_estudo` (
  `id_ciclo_estudo` int NOT NULL AUTO_INCREMENT,
  `data` date NOT NULL,
  `inicio_ciclo` time NOT NULL,
  `fim_ciclo` time NOT NULL,
  `tempo_ciclo` time DEFAULT NULL,
  `id_curso` int DEFAULT NULL,
  `id_assunto` int NOT NULL,
  PRIMARY KEY (`id_ciclo_estudo`),
  KEY `id_curso` (`id_curso`),
  KEY `id_assunto` (`id_assunto`),
  CONSTRAINT `ciclo_estudo_ibfk_1` FOREIGN KEY (`id_curso`) REFERENCES `curso` (`id_curso`),
  CONSTRAINT `ciclo_estudo_ibfk_2` FOREIGN KEY (`id_assunto`) REFERENCES `assunto` (`id_assunto`)
);

alter table curso add column turno varchar(20);
```


Inserts

São comandos que permitem que seja inserido dados nas tabelas.



```
-- DML
-- Area de Conhecimento
INSERT INTO `area_conhecimento` VALUES (8,'Humanas'),
(9,'Exatas'),
(10,'Biológicas'),
(11,'Químicas'),
(12,'Físicas'),
(13,'espaciais'),
(14,'Tecnologia da informação'),
(15,'Comunicação'),
(16,'Designer'),
(17,'Beleza e estética');

-- Nivel Curso
INSERT INTO `nivel_curso` VALUES (1,'Fundamental'),
(2,'Medio'),
(3,'Superior'),
(4,'Especialização'),
(5,'Mestrado'),
(6,'Doutorado');
```

-- Disciplina

```
INSERT INTO `disciplina` VALUES
(1,'Língua Portuguesa',15),
(2,'Tecnologia da Informação',14),
(3,'Microbiologia',10),
(4,'Raciocínio Lógico',9),
(5,'Matemática Básica',9),
(6,'Matemática Avançada',9),
(7,'Direito Constitucional',8);
```

-- Assunto

```
insert into assunto(nome, descricao, id_disciplina) values
('Lógica de primeira ordem!', 'quantificadores',4),
('Direitos e deveres individuais e coletivos', 'Direitos Políticos',7),
('Princípios Fundamentais da Contagem', 'Arranjos',5);
```

```
insert into assunto(nome, descricao, id_disciplina) values ('Programação Java', 'Java Collections',2);
```

-- Curso

```
insert into curso(nome, carga_horaria, inicio, previsao_conclusao, status, id_nivel_curso, turno) values
('Desenvolvimento-Web', 250, '2024-01-11', '2024-05-11', 'Ativo', 3, 'livre');
```

```
insert into curso(nome, carga_horaria, inicio, previsao_conclusao, status, id_nivel_curso, turno) values
('Relações Humanas', 250, '2024-02-11', '2024-06-11', 'Ativo', 3, 'tarde');
```

-- Pessoa

```
insert into pessoa(nome, rg, cpf, email, sexo, nascimento) values
('Marcos Franco','5552330-DF','22533366689','francomarcos@hotmail.com','masculino','1995-02-23'),
('Thaís Leal','2252330-MG','38533366689','tataleal@hotmail.com','feminino','1999-03-23'),
('Rafael Dourado','7766330-MG','77633366689','rafaeldourado@gmail.com','masculino','2005-03-23'),
('Roberto Santiago','20272330-BA','28533366614','robero101@hotmail.com','masculino','2002-03-30');
```

-- Professor

```
insert into professor(formacao, tipo_contrato, salario, id_pessoa) values
('Cientista de Dados', 'CLT-Indeterminado', 15000, 1),
('Letras Inglês e Português', 'CLT-Indeterminado', 9000, 2);
```

```
insert into endereco(cep, pais, uf, municipio, bairro, rua, numero, complemento, id_pessoa) values
('712530', 'Brasil', 'DF', 'Brasília', 'Asa Sul', '12', 133, 'casa',1),
('125341', 'Brasil', 'BA', 'Brasília', 'Asa Norte', '15', 233, 'casa',2),
('812532', 'Brasil', 'MG', 'Belo Horizonte', 'Centro', '12', 133, 'casa',3),
('412535', 'Brasil', 'RJ', 'Rio de Janeiro', 'Copacabana', '12', 133, 'casa',4);
```

-- Aluno

```
insert into aluno(matricula, id_pessoa) values ('GC-20245830', 3), ('GC-20245831', 4);
```

```
insert into telefone(numero, id_pessoa) values ('6199999999', 1), ('6199999988', 2), ('6199999977', 3), ('6199999966', 4);
```

Tabelas associativas

As tabelas associativas surgem a partir do relacionamento de cardinalidade, muito para muitos (m,m), já que esse tipo de relacionamento não pode ser representado apenas com as chaves das respectivas tabelas. Elas servem como ponte de ligação entre duas entidades. Neste exemplo existem 3: aluno_curso, professor_disciplina e disciplina área conhecimento. Abaixo, observa-se a utilização dessas tabelas por meio dos **inserts e selects** (comandos de seleção) combinados com os **joins** (comandos de combinação).

```

218 -----
219 -- Tabelas associativas (aluno curso
220 -- Aluno Curso
221 -- Rafael Dourado foi matriculado no curso de Des. Web
222 • insert into aluno_curso(id_aluno, id_curso) values (1,1);
223 -- Roberto Santiago foi matriculado no curso de Rel Humanas
224 • insert into aluno_curso(id_aluno, id_curso) values (2,2);
225 -- A seleção abaixo demonstra isso:
226 • select p.nome as nome_aluno, c.nome as curso_matriculado
227 from aluno a
228 inner join pessoa p on p.id_pessoa = a.id_pessoa
229 inner join aluno_curso ac on ac.id_aluno = a.id_aluno
230 inner join curso c on c.id_curso = ac.id_curso;

```



Result Grid



Filter Rows:




Export:



Wrap Cell Content:

	nome_aluno	curso_matriculado
▶	Rafael Dourado	Desenvolvimento-Web
	Roberto Santiago	Relações Humanas

```
inner join curso c on c.id_curso = ac.id_curso;
-- DQL
-- Inner Join -- Selecionar nome do aluno e matrícula.
select p.nome as nome_aluno, a.matricula, a.id_aluno
from aluno a
inner join pessoa p on p.id_pessoa = a.id_pessoa;
```

Result Grid			Filter Rows: <input type="text"/>	Export: 
	nome_aluno	matricula	id_aluno	
▶	Rafael Dourado	GC-20245830	1	
	Roberto Santiago	GC-20245831	2	


```
238      -- Professor Disciplina
239 •   insert into professor_disciplina(id_professor, id_disciplina) values (1,2),
240      (2,1);
241      -- Percebe-se que Marcos Franco leciona TI e Thaisa Leciona L. Portuguesa
242 •   select p.nome as nome_professor, d.nome as nome_disciplina
243
244      from professor prof
245      inner join professor_disciplina pd on pd.id_professor = prof.id_professor
246      inner join disciplina d on d.id_disciplina = pd.id_disciplina
247      inner join pessoa p on prof.id_pessoa = p.id_pessoa;
248
```

<   Filter Rows: | Export:  | Wrap Cell Content: 

	nome_professor	nome_disciplina
▶	Marcos Franco	Tecnologia da Informação
	Thaisa Leal	Lingua Portuguesa


```

251 -- Disciplinas cadastradas no curso
252 insert into curso_disciplina(id_curso, id_disciplina) values (1,2), (1,5);
253
254 -- Percebe-se duas disciplinas cadastradas no curso de Des. Web:
255
256 • select c.nome as nome_curso, d.nome as nome_disciplina
257
258 from curso c
259 inner join curso_disciplina cd on cd.id_curso = c.id_curso
260 inner join disciplina d on d.id_disciplina = cd.id_disciplina;

```



Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	nome_curso	nome_disciplina
▶	Desenvolvimento-Web	Tecnologia da Informação
	Desenvolvimento-Web	Matemática Básica

Ponto Final do BD.

Finaliza-se o bd Ciclos de Estudo com a implementação e uso da entidade `ciclo_estudo`, assim representa-se o uso da regra de negócio para registrar ciclos de estudo.



```
-- Ciclo de Estudo
-- Por fim o topico principal do escopo do projeto: Cadastrar ciclos de estudo
-- no insert não é necessário cadastrar o tempo do ciclo pois ele foi automatizado pela trigger abaixo:
DELIMITER //
CREATE TRIGGER subtracao_tempo
BEFORE INSERT ON ciclo_estudo
FOR EACH ROW
BEGIN
    SET NEW.tempo_ciclo = NEW.fim_ciclo - NEW.inicio_ciclo;
END;
//
```

--

```
insert into ciclo_estudo(data, inicio_ciclo, fim_ciclo, id_curso, id_assunto) values  
( '2023-12-15', '09:30:00', '09:58:00', 1, 1),  
( '2023-12-15', '09:30:00', '09:58:00', 1, 1),  
( '2023-12-15', '08:30:00', '09:10:00', 2, 2);
```

-- O exemplo de junção abaixo seleciona o assunto de acordo com a disciplina, o curso e mostra o tempo
-- gasto durante o ciclo de estudo.

```
select d.nome as diciplina, ass.nome as assunto_estudado, c.nome as curso, ce.tempo_ciclo as tempo_estudo  
  
from ciclo_estudo ce  
inner join assunto ass on ass.id_assunto = ce.id_assunto  
inner join curso c on c.id_curso = ce.id_curso  
inner join disciplina d on d.id_disciplina = ass.id_disciplina;
```

Conclusão

Portanto é fundamental aplicar, durante o desenvolvimento, todas as fases do banco de dados, desde a análise de requisitos à implementação. Destaca-se a importância da modelagem e diagramação no desenvolvimento dos bancos de dados.
