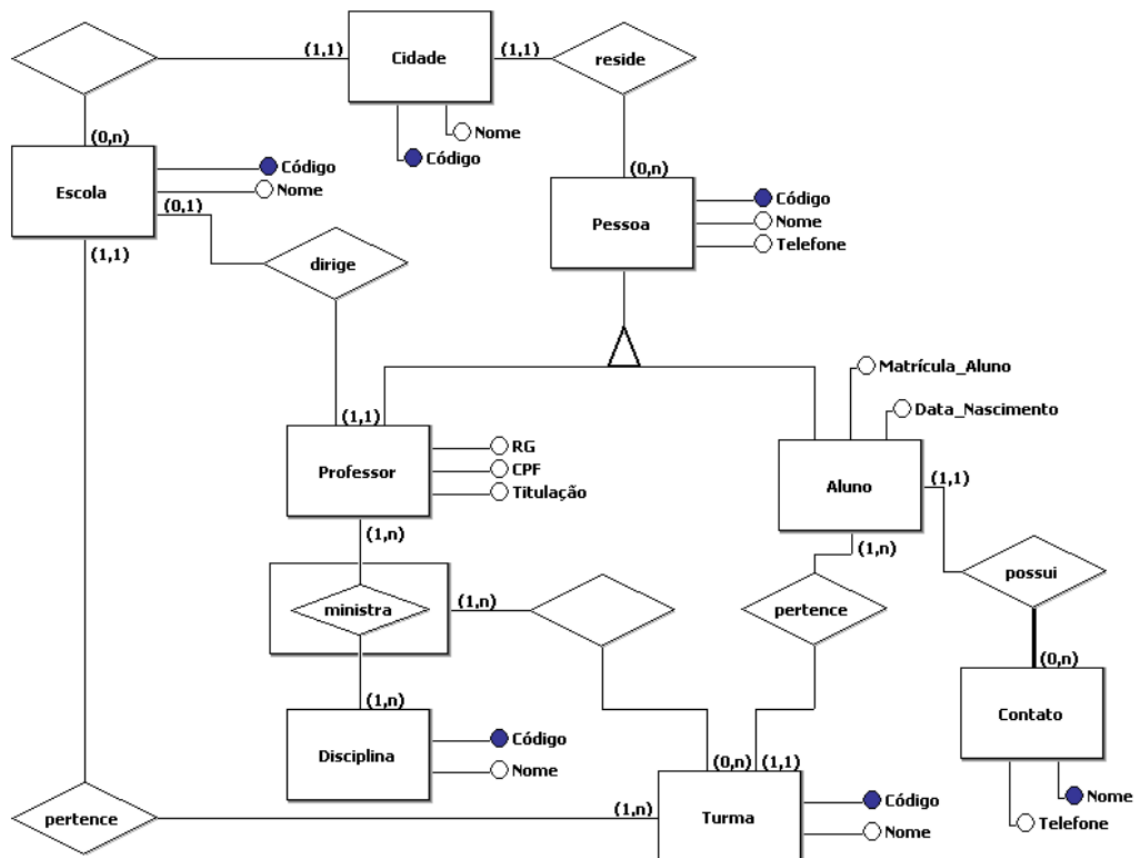


# TRABALHO BD

Alunos: Romulo, Jhonatan e Suelen

Modelo Conceitual

## Modelo Conceitual: Escola



## Modelo Lógico

Cidade (CodCidade, Nome)

Escola (CodEscola, Nome, CodDiretor, CodCidade)  
CodDiretor referencia Professor  
CodCidade referencia Cidade

Turma (CodTurma, CodEscola, Nome)  
CodEscola referencia Escola

Pessoa (CodPessoa, CodCidade, Nome, Telefone)  
CodCidade referencia Cidade

Professor (CodPessoa, RG, CPF, Titulacao)  
CodPessoa referencia Pessoa

Aluno (CodPessoa, Matricula, Data\_Nasc, CodTurma)  
CodPessoa referencia Pessoa  
CodTurma referencia Turma

Disciplina (CodDisciplina, Nome)

Professor\_Disciplina (CodDisciplina, CodPessoa)  
CodDisciplina referencia Disciplina  
CodPessoa referencia Professor

Ministra\_Turma (CodDisciplina, CodPessoa, CodTurma)  
CodDisciplina, CodPessoa referencia Professor\_Disciplina  
CodTurma referencia Turma

Contato (CodPessoa, Nome, Telefone)  
CodPessoa referencia Aluno

## Creates

```
CREATE TABLE cidade (  
    cidade_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    nome VARCHAR(45) NOT NULL,  
    CONSTRAINT pk_cidade PRIMARY KEY(cidade_id)  
);  
  
CREATE TABLE disciplina(  
    disciplina_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    nome VARCHAR(45),  
    CONSTRAINT pk_disciplina PRIMARY KEY (disciplina_id)  
);  
  
CREATE TABLE pessoa (  
    pessoa_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,  
    cidade_id SMALLINT UNSIGNED NOT NULL,  
    nome VARCHAR(45) NOT NULL,  
    telefone VARCHAR(45),  
    CONSTRAINT pk_pessoa PRIMARY KEY (pessoa_id),  
    CONSTRAINT fk_pessoa_cidade FOREIGN KEY (cidade_id)  
    REFERENCES cidade (cidade_id)  
    ON DELETE RESTRICT ON UPDATE CASCADE  
);  
  
CREATE TABLE professor(  
    professor_id SMALLINT UNSIGNED NOT NULL,  
    rg VARCHAR(10) NOT NULL,  
    cpf VARCHAR(11) NOT NULL,  
    titulacao VARCHAR(45),  
    CONSTRAINT uk_professor_cpf UNIQUE KEY(cpf),  
    CONSTRAINT uk_professor_rg UNIQUE KEY(rg),  
    CONSTRAINT pk_professor PRIMARY KEY (professor_id),  
    CONSTRAINT fk_professor_pessoa FOREIGN KEY (professor_id)  
    REFERENCES pessoa (pessoa_id)  
    ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```

CREATE TABLE escola(
    escola_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
    nome VARCHAR(50),
    diretor_id SMALLINT UNSIGNED NOT NULL,
    cidade_id SMALLINT UNSIGNED NOT NULL,
    CONSTRAINT pk_escola PRIMARY KEY(escola_id),
    CONSTRAINT uk_escola_professor UNIQUE KEY (diretor_id),
    CONSTRAINT fk_escola_professor FOREIGN KEY (diretor_id)
    REFERENCES professor (professor_id)
    ON DELETE RESTRICT ON UPDATE CASCADE,
    CONSTRAINT fk_escola_cidade FOREIGN KEY (cidade_id)
    REFERENCES cidade (cidade_id)
    ON DELETE RESTRICT ON UPDATE CASCADE
);

```

```

CREATE TABLE turma(
    turma_id SMALLINT UNSIGNED NOT NULL AUTO_INCREMENT,
    escola_id SMALLINT UNSIGNED NOT NULL,
    nome VARCHAR(45) NOT NULL,
    CONSTRAINT pk_turma PRIMARY KEY (turma_id),
    CONSTRAINT fk_turma_escola FOREIGN KEY (escola_id)
    REFERENCES escola (escola_id)
    ON DELETE RESTRICT ON UPDATE CASCADE
);

```

```

CREATE TABLE aluno(
    aluno_id SMALLINT UNSIGNED NOT NULL,
    turma_id SMALLINT UNSIGNED NOT NULL,
    matricula INT NOT NULL,
    data_nasc DATE NOT NULL,
    CONSTRAINT pk_aluno PRIMARY KEY (aluno_id),
    CONSTRAINT uk_aluno_pessoa UNIQUE KEY (matricula),
    CONSTRAINT fk_aluno_pessoa FOREIGN KEY (aluno_id)
    REFERENCES pessoa (pessoa_id)
    ON DELETE RESTRICT ON UPDATE CASCADE,
    CONSTRAINT fk_aluno_turma FOREIGN KEY (turma_id)
    REFERENCES turma (turma_id)
    ON DELETE RESTRICT ON UPDATE CASCADE
);

```

```

CREATE TABLE contato(
    aluno_id SMALLINT UNSIGNED NOT NULL,
    nome VARCHAR(45) NOT NULL,
    telefone VARCHAR(45) NOT NULL,
    CONSTRAINT pk_contato PRIMARY KEY (aluno_id,nome),
    CONSTRAINT fk_aluno_contato FOREIGN KEY (aluno_id)
    REFERENCES aluno (aluno_id)
    ON DELETE RESTRICT ON UPDATE CASCADE
);

CREATE TABLE professor_disciplina(
    disciplina_id SMALLINT UNSIGNED NOT NULL,
    professor_id SMALLINT UNSIGNED NOT NULL,
    CONSTRAINT pk_professor_disciplina PRIMARY KEY (disciplina_id, professor_id),
    CONSTRAINT fk_professor_disciplina_disciplina FOREIGN KEY (disciplina_id)
    REFERENCES disciplina (disciplina_id)
    ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_professor_disciplina_professor FOREIGN KEY (professor_id)
    REFERENCES professor (professor_id)
    ON DELETE CASCADE ON UPDATE CASCADE
);

CREATE TABLE ministra_turma(
    disciplina_id SMALLINT UNSIGNED NOT NULL,
    professor_id SMALLINT UNSIGNED NOT NULL,
    turma_id SMALLINT UNSIGNED NOT NULL,
    CONSTRAINT pk_ministra_turma PRIMARY KEY (disciplina_id, professor_id,turma_id),
    CONSTRAINT fk_ministra_turma_professor_disciplina
    FOREIGN KEY (disciplina_id, professor_id)
    REFERENCES professor_disciplina (disciplina_id, professor_id)
    ON DELETE CASCADE ON UPDATE CASCADE,
    CONSTRAINT fk_ministra_turma_turma FOREIGN KEY (turma_id)
    REFERENCES turma (turma_id)
    ON DELETE CASCADE ON UPDATE CASCADE
);

```

## Inserts

### #Cidades

```
INSERT INTO cidade(nome) VALUES ("Rio de Janeiro"),
("Niterói"), ("Volta Redonda"),
("Rio das Ostras"), ("Resende"),
("Maricá");
```

### #Disciplinas

```
INSERT INTO disciplina(nome) VALUES ("Matemática"),("Inglês"),("Química"),("Espanhol"),
("Biologia"),("Filosofia"),("Educação Física"),
("Geografia"),("História da América"),
("História do Brasil"),("História Geral"),
("Geografia do Brasil"),("Português"),
("Literatura"),("Física"),("Sociologia "),
("Redação"),("Artes");
```

### #Pessoas

```
INSERT INTO pessoa(cidade_id, nome, telefone) VALUES
( 1, "Daniel Melo", "(21) 6502-4376"), ( 1, "Ana Silva", "(21) 7650-8595"),
( 1, "Carolina Pinto", "(21) 4940-9643"),
( 1, "Gabriel Oliveira", "(21) 3028-5919"), ( 1, "Nicole Rocha", "(21) 4087-4927"),
( 1, "Kauê Cavalcanti", "(21) 2900-2656"), ( 1, "Ryan Costa", "(21) 6495-5737"),
( 1, "André Rocha", "(21) 5663-3216"), ( 2, "Luís Almeida", "(21) 2282-5728"),
( 2, "Emily Rodrigues", "(21) 9219-6557"), ( 2, "Larissa Silva", "(21) 3636-7527"),
( 2, "Samuel Costa", "(21) 3952-2772"),
( 2, "Rodrigo Oliveira", "(21) 6214-5729"), ( 4, "Felipe Costa", "(21) 2326-5774"),
( 3, "Tiago Oliveira", "(21) 4976-5741"),
( 3, "Lavinia Carvalho", "(21) 5223-9794"), ( 3, "Bianca Souza", "(21) 5033-7598"),
( 3, "Vinicius Barbosa", "(21) 5154-3647"), ( 3, "José Pinto", "(21) 2025-7722"),
( 3, "Renan Oliveira", "(21) 4321-3996"), ( 4, "Larissa Cardoso", "(21) 2261-8608"),
( 4, "Julian Cavalcanti", "(21) 5193-5808"), ( 4, "Evelyn Dias", "(21) 7453-5519"),
( 4, "Alex Pinto", "(21) 2135-2523"), ( 4, "Igor Araujo", "(21) 5996-5190"),
( 4, "Larissa Pinto", "(21) 7122-3031"), ( 4, "Breno Ferreira", "(21) 2434-4966"),
( 4, "Estevan Barbosa", "(21) 2277-7809"), ( 5, "Felipe Pereira", "(21) 3082-2715"),
( 5, "Gabrielle Almeida", "(21) 8370-4073"), ( 5, "Evelyn Sousa", "(21) 5906-4076"),
( 5, "Marcos Carvalho", "(21) 3600-4680"),
( 5, "Ryan Costa", "(21) 6790-4138"), ( 5, "Letícia Alves", "(21) 7185-9449"),
( 5, "Leila Castro", "(21) 8225-6842"), ( 5, "Fernanda Costa", "(21) 6522-4263"),
( 5, "Victor Lima", "(21) 3482-9220"), ( 5, "Felipe Gomes", "(21) 5574-7364"),
( 5, "Matheus Silva", "(21) 5990-6045"), ( 6, "Kauê Ferreira", "(21) 6839-7328"),
( 6, "Miguel Pinto", "(21) 6953-8260"), ( 6, "Diego Azevedo", "(21) 8471-5922"),
( 6, "Davi Dias", "(21) 8466-7501"), ( 6, "Cauã Goncalves", "(21) 4170-6206"),
( 6, "Giovanna Rodrigues", "(21) 8327-7153"),
( 6, "Kaua Cunha", "(21) 3531-6756"), ( 6, "Murilo Rodrigues", "(21) 4571-4774"),
( 6, "Gabriel Alves", "(21) 7406-6346"),
( 6, "Gustavo Melo", "(21) 7844-6513"), ( 6, "Matheus Rocha", "(21) 4125-5538"),
( 3, "José Mendes", "(21) 2357-4476"),
( 6, "Alessandro Pimentel", "(21) 8632-4366");
```

```

#Professor
INSERT INTO professor(professor_id, rg, cpf, titulacao) VALUES
(1,"20.821.009-0" ,"102.223.536-03","Mestre"),
(2,"13.820.793-5" ,"129.324.524-42","Especialista"),
(3,"12.790.347-6" ,"093.399.249-16","Mestre"),
(4,"14.091.510-6" ,"100.769.979-52","Mestre"),
(5,"13.899.943-2" ,"109.287.639-19","Especialista"),
(6,"12.790.350-6" ,"093.399-259-98","Especialista"),
(7,"13.856.023-6" ,"021.052.776-59",null),
(8,"13.951.736-9" ,"110.219.359-30","Especialista"),
(9,"13.821.069-3" ,"107.777.819-80","Mestre"),
(10,"10.282.018-5" ,"102.347.829-50","Doutor"),
(11,"24.362.016-1" ,"090.390.819-13","Especialista"),
(12,"17.488.058-0" ,"101.447.907-10","Doutor");

#Escola
INSERT INTO escola(nome, diretor_id, cidade_id) VALUES
("ESCOLA MUNICIPAL JORNALISTA FREITAS NETO",1,1),
("ESCOLA MUNICIPAL ROMEU DE AVELAR",2,1),
("ESCOLA MUNICIPAL MARIO DAVID ANDREAZZA",3,2),
("ESCOLA MUNICIPAL PROF. RUTH DE ALMEIDA BEZERRA",4,3),
("ESCOLA MUNICIPAL SERAFINI COSTAPERARIA",5,4);

#Turma
INSERT INTO turma(escola_id, nome) VALUES
(1,"A201"), (2,"C203"), (1, "A205"),
(3,"B402"), (4,"V301"), (5,"L406");

#Aluno
INSERT INTO aluno(aluno_id, turma_id, matricula, data_nasc) VALUES
(13,1,50243,"1991-01-09"),(14,6,65085,"1992-06-17"),
(15,1,94096,"1992-12-16"),(16,1,32859,"1993-07-07"),
(17,2,28749,"1993-09-22"),(18,2,90026,"1993-11-17"),
(19,2,49557,"1994-05-18"),(20,2,66332,"1995-08-23"),
(21,2,28257,"1995-11-29"),(22,2,21965,"1996-05-01"),
(23,3,63675,"1996-12-04"),(24,3,95227,"1997-02-12"),
(25,3,21457,"1997-04-16"),(26,3,32657,"1997-10-22"),
(27,3,97657,"1998-01-28"),(28,3,22397,"1998-03-04"),
(29,3,13375,"1998-04-01"),(30,3,15436,"1998-12-23"),
(31,3,62577,"1999-01-06"),(32,3,32139,"2000-03-22"),
(33,4,26186,"2000-05-24"),(34,4,19358,"2001-05-23"),
(35,4,45355,"2002-09-11"),(36,4,13525,"2002-11-27"),
(37,4,99651,"2002-12-11"),(38,4,12230,"1996-05-24"),
(39,4,43449,"1996-10-19"),(40,4,27778,"1997-01-28"),
(41,4,90640,"1998-02-28"),(42,4,60046,"1998-04-20"),
(43,4,79041,"1998-06-25"),(44,4,18594,"1999-02-23"),
(45,4,22568,"2000-07-03"),(46,4,52242,"2000-09-29"),
(47,4,48292,"2000-10-11"),(48,4,57473,"2001-10-22"),
(49,4,88227,"2001-12-03"),(50,4,37040,"2002-11-25"),
(51,5,62001,"1998-12-02"),(52,5,62040,"2001-03-23");

```

```

#Contato
INSERT INTO contato(aluno_id, nome, telefone) VALUES
(15,"Roberto Azevedo","(21) 5574-7364"),
(15,"Maria Conceição Pires","(21) 5990-6045"),
(16,"Lucas Martins de Oliveira","(21) 6839-7328"),
(20,"José Mendes da Costa","(21) 6953-8260"),
(30,"Fernanda Moura","(21) 8471-5922"),
(48,"Newton Nunes Xavier","(21) 8466-7501");

#Professor_Disciplina
INSERT INTO professor_disciplina(professor_id, disciplina_id) VALUES
(1,1), (12,1), (2,2), (3,18), (4,7), (5,8), (5,12), (6,3), (7,4), (8,6),
(8,16), (9,5), (10,9), (10,10), (10,11), (10,8), (10,12), (12,15);

#Ministra_Turma
INSERT INTO ministra_turma(professor_id, disciplina_id, turma_id) VALUES
(1,1,1), (1,1,2), (1,1,3), (1,1,4), (2,2,2), (3,18,2), (4,7,3),
(5,8,4), (5,8,1), (5,12,1), (6,3,2), (7,4,2), (8,6,2), (8,16,2), (9,5,2),
(10,9,4), (10,10,4), (10,11,4), (10,8,4);

```



## Consultas

- 1) Listar o nome e a cidade das escolas onde todos os alunos residam na mesma cidade onde a escola está localizada.

```
SELECT e.nome Escola, c.nome Cidade
FROM cidade c, escola e
WHERE NOT EXISTS(
    SELECT 1
    FROM aluno a, turma t, pessoa p
    WHERE a.turma_id = t.turma_id AND a.aluno_id = p.pessoa_id
        AND t.escola_id = e.escola_id AND e.cidade_id != p.cidade_id
    ) AND e.cidade_id = c.cidade_id
```

Escola	Cidade
ESCOLA MUNICIPAL SERAFINI COSTA	PERARIA
Rio das Ostras	

- 2) Listar o nome e matrícula dos alunos sem nenhum contato cadastrado.

```
SELECT nome, matricula
FROM pessoa p, aluno a
WHERE matricula NOT IN (
    SELECT matricula
    FROM aluno a, contato c
    WHERE a.aluno_id = c.aluno_id
) AND pessoa_id = aluno_id
```

nome	matricula
Felipe Gomes	12230
Felipe Pereira	13375
Fernanda Costa	13525
Cauã Goncalves	18594
Letícia Alves	19358
Igor Araujo	21457
Julian Cavalcanti	21965
Estevan Barbosa	22397
Giovanna Rodrigues	22568
Ryan Costa	26186
Kauê Ferreira	27778
Larissa Cardoso	28257
Bianca Souza	28749
Marcos Carvalho	32139
Larissa Pinto	32657
Matheus Rocha	37040
Matheus Silva	43449
Leila Castro	45355
Murilo Rodrigues	48292

nome	matricula
José Pinto	49557
Rodrigo Oliveira	50243
Kaua Cunha	52242
Diego Azevedo	60046
José Mendes	62001
Alessandro Pimentel	62040
Evelyn Sousa	62577
Evelyn Dias	63675
Felipe Costa	65085
Davi Dias	79041
Gustavo Melo	88227
Vinicius Barbosa	90026
Miguel Pinto	90640
Alex Pinto	95227
Breno Ferreira	97657
Victor Lima	99651

- 3) Listar o código e nome das turmas com mais de 5 alunos.

```
SELECT t.turma_id, t.nome
FROM aluno a, turma t
WHERE a.turma_id = t.turma_id
GROUP BY t.nome
HAVING COUNT(a.matricula) > 5
```

turma_id	nome
3	A205
4	B402
2	C203

- 4) Listar o código, nome e titulação dos professores que ministram aulas para pelo menos três turmas diferentes.

```
SELECT p.professor_id, nome, p.titulacao
FROM ministra_turma m, professor p, pessoa
WHERE p.professor_id = pessoa_id AND p.professor_id = m.professor_id
GROUP BY p.professor_id
HAVING COUNT(DISTINCT m.turma_id) >= 3
```

professor_id	nome	titulacao
1	Daniel Melo	Mestre

- 5) Listar por disciplina o número de professores que podem ministrá-la e quantos efetivamente ministram a mesma para uma turma.

```
SELECT d.nome disciplina, COUNT(distinct p.professor_id) podem_ministrar,
COUNT(distinct mt.professor_id) ministrando
FROM disciplina d
LEFT JOIN professor_disciplina pd ON d.disciplina_id = pd.disciplina_id
LEFT JOIN professor p ON pd.professor_id = p.professor_id
LEFT JOIN ministra_turma mt ON d.disciplina_id = mt.disciplina_id
GROUP BY d.nome;
```

disciplina	podem_ministrar	ministrando
Artes	1	1
Biologia	1	1
Educação Física	1	1
Espanhol	1	1
Filosofia	1	1
Física	1	0
Geografia	2	2
Geografia do Brasil	2	1
História da América	1	1
História do Brasil	1	1
História Geral	1	1
Inglês	1	1
Literatura	0	0
Matemática	2	1
Português	0	0
Química	1	1
Redação	0	0
Sociologia	1	1

- 6) Listar o nome da escola e o nome dos diretores de escola que residem em cidades diferentes da cidade da escola.

```
SELECT e.nome nome_escola, pe.nome diretor
FROM pessoa pe, escola e, professor p
WHERE e.diretor_id = p.professor_id
AND pe.pessoa_id = p.professor_id
AND e.cidade_id != pe.cidade_id
```

nome_escola	diretor
ESCOLA MUNICIPAL MARIO DAVID ANDREAZZA	Carolina Pinto
ESCOLA MUNICIPAL PROF. RUTH DE ALMEIDA BEZERRA	Gabriel Oliveira
ESCOLA MUNICIPAL SERAFINI COSTAPERARIA	Nicole Rocha

- 7) Listar por escola o número de turmas e o número de professores que ministram alguma disciplina para turmas da escola em questão.

```
SELECT e.nome, COUNT(t.turma_id) numero_de_turmas, temp.professores_ministrando
FROM turma t JOIN escola e ON t.escola_id = e.escola_id JOIN (
    SELECT e.escola_id id_da_escola,
    COUNT(DISTINCT p.professor_id) professores_ministrando
    FROM professor p
    JOIN ministra_turma mt
    ON p.professor_id = mt.professor_id
    JOIN turma t
    ON t.turma_id = mt.turma_id
    JOIN escola e
    ON e.escola_id = t.escola_id
    GROUP BY id_da_escola) temp
ON id_da_escola = e.escola_id
GROUP BY e.nome
```

nome	numero_de_turmas	professores_ministrando
ESCOLA MUNICIPAL JORNALISTA FREITAS NETO	2	3
ESCOLA MUNICIPAL MARIO DAVID ANDREAZZA	1	3
ESCOLA MUNICIPAL ROMEU DE AVELAR	1	7

- 8) Listar por escola a razão entre o número de alunos da escola e o número de professores que ministram alguma disciplina na escola em questão.

```
SELECT e.nome, ( temp.qtd_alunos / COUNT(DISTINCT p.professor_id) ) razao_prof_aluno
FROM professor p JOIN ministra_turma mt
ON p.professor_id = mt.professor_id
JOIN turma t
ON t.turma_id = mt.turma_id
JOIN escola e
ON e.escola_id = t.escola_id
JOIN (SELECT esc.escola_id, COUNT(DISTINCT a.aluno_id) qtd_alunos
FROM aluno a JOIN turma t
ON t.turma_id = a.turma_id
JOIN escola esc
ON esc.escola_id = t.escola_id
GROUP BY esc.escola_id) temp
ON temp.escola_id = e.escola_id
GROUP BY e.escola_id
```

nome	razao_prof_aluno
ESCOLA MUNICIPAL JORNALISTA FREITAS NETO	4.3333
ESCOLA MUNICIPAL ROMEU DE AVELAR	0.8571
ESCOLA MUNICIPAL MARIO DAVID ANDREAZZA	6.0000

- 9) Listar todos os contatos dos alunos informando a matrícula e nome do aluno, nome e telefone do contato, ordenado por matrícula do aluno e nome do contato.

```
SELECT a.matricula, p.nome aluno, p.nome nome_do_contato,
c.telefone telefone_do_contato
FROM pessoa p, aluno a, contato c
WHERE a.aluno_id = c.aluno_id AND p.pessoa_id = a.aluno_id
ORDER BY a.matricula, nome_do_contato
```

matricula	aluno	nome_do_contato	telefone_do_contato
15436	Gabrielle Almeida	Fernanda Moura	(21) 8471-5922
32859	Lavinia Carvalho	Lucas Martins de Oliveira	(21) 6839-7328
57473	Gabriel Alves	Newton Nunes Xavier	(21) 8466-7501
66332	Renan Oliveira	José Mendes da Costa	(21) 6953-8260
94096	Tiago Oliveira	Maria Conceição Pires	(21) 5990-6045
94096	Tiago Oliveira	Roberto Azevedo	(21) 5574-7364

- 10) Listar todos os professores que ministram disciplinas para apenas uma turma.

```
SELECT pessoa.nome
FROM pessoa, professor p, ministra_turma m
WHERE p.professor_id = m.professor_id
AND pessoa.pessoa_id = p.professor_id
GROUP BY pessoa.nome
HAVING COUNT(DISTINCT m.turma_id) = 1
```

nome

Ana Silva

André Rocha

Carolina Pinto

Emily Rodrigues

Gabriel Oliveira

Kauê Cavalcanti

Luís Almeida

Ryan Costa

## Updates

- 1) Alterar todos os contatos vinculados a um aluno A para um aluno B.

```
SELECT @origem := 15;
SELECT @destino := 48;
```

```
UPDATE contato c
SET c.aluno_id = @destino
WHERE c.aluno_id = @origem
```

- 2) Forneça um conjunto de instruções SQL para remover da base de dados um professor qualquer. Caso o professor seja diretor de uma escola um novo diretor deverá ser definido.

```
SELECT @professorId := 3;
SELECT @novoDiretor := 8;
```

```
/* remover professor ID = 1*/
UPDATE escola
SET diretor_id = @novoDiretor
WHERE diretor_id = @professorId;
```

```
/* remove o professor especificado*/
DELETE FROM professor
WHERE professor_id = @professorId;
```

- 3) Considerando que um Professor P1 precisou sair de licença médica, substituí-lo por outro Professor P2 em todas as turmas onde P1 ministra disciplinas. Importante: Caso o Professor P2 não esteja vinculado a todas as disciplinas ministradas pelo Professor P1, inclua a informação de que o Professor P2 é capaz de ministrar todas as disciplinas ministradas pelo Professor P1.

```
SELECT @afastado := 1;
SELECT @substituto := 5;
```

```
/* Insere as materias que não eram dadas pelo professor substituto */
INSERT IGNORE INTO professor_disciplina(disciplina_id, professor_id)
SELECT disciplina_id, @substituto
FROM professor_disciplina
WHERE professor_id = @afastado;
```

```
/* seta as turmas do professor de licença para o professor substituto*/
UPDATE ministra_turma
SET professor_id = @substituto
WHERE professor_id = @afastado
```



## Outras consultas que utilizamos ao longo do trabalho

- Quantidade de alunos por turma

```
“sql SELECT COUNT(a.matricula), t.nome FROM aluno a, turma t WHERE a.turma_id = t.turma_id  
GROUP BY t.nome
```

““

- Mostrar todos os alunos que tem contato

```
SELECT nome, matricula  
FROM pessoa,(  
  SELECT a.aluno_id, matricula  
  FROM aluno a, contato c  
  WHERE a.aluno_id = c.aluno_id  
) r  
WHERE pessoa_id = r.aluno_id
```

- Quantidade de alunos por escola

```
SELECT e.nome nome_da_escola, COUNT(DISTINCT a.aluno_id) qtd_alunos  
FROM aluno a JOIN turma t ON t.turma_id = a.turma_id  
JOIN escola e ON e.escola_id = t.escola_id  
GROUP BY nome_da_escola
```