



Teach Up - Documentação Complementar

Discente: Diego Lucas Hattori Dallaqua

RA: 2337703

Docente: Leiliane Rezende

Disciplina: CC23B

Turma: CC32

Santa Helena

2023

Consultas

-- Funções de Linha e Funções de Agregação

```
SELECT nome AS Nome_Usuario, descricao AS Genero, COUNT(*) AS Qtd_Professores  
  
FROM tb_user  
  
JOIN tb_professor ON tb_professor.user_id = tb_user.id_user  
  
JOIN tb_genero ON tb_genero.id_genero = tb_user.genero_id  
  
GROUP BY nome, descricao  
  
HAVING COUNT(*) >= 1;
```

	Nome_Usuario	Genero	Qtd_Professores
▶	Laura Roberta	Feminino	1
	Maria Eduarda	Feminino	1
	Laura Alves	Feminino	1
	Maria Fernandes	Feminino	1
	Mateus Costa	Masculino	1
	João Silva	Masculino	1
	Matheus Costa	Masculino	1
	João Sousa	Masculino	1
	Pedro Souza	Outro	1
	Pedro Santos	Outro	1

-- Junções e Operadores de Conjuntos

```
( SELECT tb_escola.nome_escola AS Nome_Escola, tb_cidade.nome AS Nome_Cidade  
  
FROM tb_escola  
  
JOIN tb_endereco ON tb_endereco.id_endereço = tb_escola.endereco_id  
  
JOIN tb_cidade ON tb_endereco.cidade_id = tb_cidade.id_cidade  
  
WHERE qtd_alunos >= 600  
  
ORDER BY tb_escola.nome_escola )  
  
UNION
```

```
( SELECT tb_escola.nome_escola AS Nome_Escola, tb_cidade.nome AS Nome_Cidade
FROM tb_escola
JOIN tb_endereco ON tb_endereco.id_endereço = tb_escola.endereco_id
JOIN tb_cidade ON tb_endereco.cidade_id = tb_cidade.id_cidade
WHERE qtd_alunos <= 400
ORDER BY tb_escola.nome_escola );
```

	Nome_Escola	Nome_Cidade
►	Escola 2	Campinas
	Escola 4	Assis
	Escola 7	Maringá
	Escola 9	Cascavel
	Escola 1	Presidente Prudente
	Escola 3	São Paulo
	Escola 8	Foz do Iguaçu

-- Sub-consultas

```
SELECT nome_escola, qtd_alunos
FROM tb_escola
WHERE qtd_alunos >= ( SELECT AVG(qtd_alunos) FROM tb_escola );
```

	nome_escola	qtd_alunos
►	Escola 2	672
	Escola 4	764
	Escola 7	871
	Escola 9	652
	Escola 10	562

Comandos especiais de inserção, atualização e remoção

Inserção

```
INSERT INTO tb_professor (user_id)

SELECT id_user

FROM tb_user

WHERE (SELECT COUNT(*) FROM tb_professor) < 1000

LIMIT 1;
```

Atualização

```
UPDATE tb_moderador

SET salario = salario * 1.25

WHERE id_moderador <= 5 AND salario < 2500;
```

	id_moderador	user_id	endereco_id	salario
▶	1	1	1	2000
	2	2	2	2500
	3	3	3	3000
	4	4	4	2500
	5	5	5	2000
	6	11	6	2000
	7	12	7	2500
	8	13	8	2500
	9	14	9	2000
	10	15	10	2500
*	NULL	NULL	NULL	NULL

	id_moderador	user_id	endereco_id	salario
▶	1	1	1	2500
	2	2	2	2500
	3	3	3	3000
	4	4	4	2500
	5	5	5	2500
	6	11	6	2000
	7	12	7	2500
	8	13	8	2500
	9	14	9	2000
	10	15	10	2500
*	NULL	NULL	NULL	NULL

Remoção

```
DELETE FROM tb_escola
```

```
WHERE id_escola < 8 AND qtd_alunos < 400;
```

View

```
CREATE VIEW relatorio_turma AS
```

```
SELECT tb_turma.sigla AS Sigla_Turma, tb_disciplina.nome AS Disciplina,  
tb_escola.nome_escola AS Nome_Escola, tb_escola.qtd_alunos AS Qtd_Alunos,  
tb_cidade.nome AS Nome_Cidade
```

```
FROM tb_turma
```

```
JOIN tb_professor ON tb_turma.professor_id = tb_professor.id_professor
```

```
JOIN tb_disciplina ON tb_turma.disciplina_id = tb_disciplina.id_disciplina
```

```
JOIN tb_escola ON tb_disciplina.escola_id = tb_escola.id_escola
```

```
JOIN tb_endereco ON tb_escola.endereco_id = tb_endereco.id_endereço
```

```
JOIN tb_cidade ON tb_endereco.cidade_id = tb_cidade.id_cidade
```

```
GROUP BY tb_turma.sigla, tb_disciplina.nome, tb_escola.nome_escola,  
tb_escola.qtd_alunos, tb_cidade.nome;
```

	Sigla_Turma	Disciplina	Nome_Escola	Qtd_Alunos	Nome_Cidade
►	1A	Educação Física	Escola 10	562	Presidente Prudente
	2B	Geografia	Escola 9	652	Cascavel
	3C	História	Escola 8	321	Foz do Iguaçu
	1D	Biologia	Escola 7	871	Maringá
	2A	Química	Escola 6	419	Londrina
	3B	Física	Escola 5	526	Santa Helena
	1C	Inglês	Escola 4	764	Assis
	2D	Espanhol	Escola 3	386	São Paulo
	3A	Português	Escola 2	672	Campinas
	1B	Matemática	Escola 1	248	Presidente Prudente

Função

DELIMITER //

```
CREATE FUNCTION aumentar_salario( aux_id_moderador INT, percentual_aumento
DECIMAL (5,2) )
RETURNS decimal(10,2) DETERMINISTIC
BEGIN
```

```
    DECLARE novo_salario, aumento DECIMAL(10,2);
```

```
    -- Obter o salário atual do moderador
    SELECT salario INTO novo_salario
        FROM tb_moderador
        WHERE id_moderador = aux_id_moderador;
```

```
    -- Verificar se o moderador existe
    IF novo_salario IS NOT NULL THEN
```

```
        -- Calcular o aumento com base no percentual
        SET aumento = novo_salario * ( percentual_aumento / 100 );
```

```
        -- Atualizar o salário com o aumento
        UPDATE tb_moderador
            SET salario = novo_salario + aumento
            WHERE id_moderador = aux_id_moderador;
```

```
        -- Retornar o novo salário
        RETURN novo_salario + aumento;
```

```
    ELSE
```

```
        -- Retornar -1 se o moderador não existir
        RETURN -1.00;
```

```
    END IF;
END //
```

DELIMITER ;

```
SELECT aumentar_salario ( 1, 50.0 );
```

	id_moderador	user_id	endereco_id	salario
▶	1	1	1	2500
	2	2	2	2500
	3	3	3	3000
	4	4	4	2500
	5	5	5	2500
	6	11	6	2000
	7	12	7	2500
	8	13	8	2500
	9	14	9	2000
	10	15	10	2500
•	NULL	NULL	NULL	NULL

	id_moderador	user_id	endereco_id	salario
▶	1	1	1	3750
	2	2	2	2500
	3	3	3	3000
	4	4	4	2500
	5	5	5	2500
	6	11	6	2000
	7	12	7	2500
	8	13	8	2500
	9	14	9	2000
	10	15	10	2500
•	NULL	NULL	NULL	NULL

Procedimentos

DELIMITER //

```
CREATE PROCEDURE proc_aumentar_salario( aux_id_moderador INT,
percentual_aumento DECIMAL(5,2) )
BEGIN
```

```
    DECLARE novo_salario, aumento DECIMAL(10,2);
```

```
    -- Obtenha o salário atual do moderador
```

```
    SELECT salario INTO novo_salario
```

```
        FROM tb_moderador
```

```
        WHERE id_moderador = aux_id_moderador;
```

```
    -- Aumente o salário usando o percentual
```

```
    SET aumento = novo_salario * ( percentual_aumento / 100 );
```

```
    -- Atualize o salário na tabela
```

```
    UPDATE tb_moderador
```

```
        SET salario = novo_salario + aumento
```

```
        WHERE id_moderador = aux_id_moderador;
```

```
END //
```

```
DELIMITER ;
```

```
CALL proc_aumentar_salario( 6, 50.0 );
```

	id_moderador	user_id	endereco_id	salario
▶	1	1	1	3750
	2	2	2	2500
	3	3	3	3000
	4	4	4	2500
	5	5	5	3750
	6	11	6	2250
	7	12	7	2500
	8	13	8	2500
	9	14	9	2000
	10	15	10	2500
★	NULL	NULL	NULL	NULL

	id_moderador	user_id	endereco_id	salario
▶	1	1	1	3750
	2	2	2	2500
	3	3	3	3000
	4	4	4	2500
	5	5	5	3750
	6	11	6	3375
	7	12	7	2500
	8	13	8	2500
	9	14	9	2000
	10	15	10	2500
★	NULL	NULL	NULL	NULL

Evento

DELIMITER //

```
CREATE EVENT aniversario
ON SCHEDULE EVERY 1 DAY
DO
```

```
BEGIN
```

```
    DECLARE today DATE;
```

```
    DECLARE user_id INT;
```

```
    SET today = CURDATE();
```

```
    -- Obter usuários cujo aniversário é hoje
```

```
    SELECT id_user INTO user_id
```

```
        FROM tb_user
```

```
        WHERE DAYOFMONTH(id_user.dt_nascimento) = DAYOFMONTH(today)
```

```
        AND MONTH(id_user.dt_nascimento) = MONTH(today)
```

```
        LIMIT 1;
```

```
    -- Verificar se há usuários com aniversário hoje
```

```
    IF user_id IS NOT NULL THEN
```

```
        -- Enviar mensagem ou executar ação desejada
```

```
        INSERT INTO tb_mensagem
```

```
            VALUES (user_id, 'Feliz Aniversário!', NOW());
```

```
    END IF;
```

```
END //
```

DELIMITER ;

Trigger

DELIMITER //

```
CREATE TRIGGER cadastro
AFTER INSERT ON tb_user
FOR EACH ROW
BEGIN
    DECLARE user_id INT;

    -- Obter o ID do usuário recém-cadastrado
    SELECT id_user INTO user_id
        FROM tb_user
        WHERE id_user = NEW.id_user;

    -- Exibir mensagem no console (substitua pelo código desejado)
    INSERT INTO tb_mensagem
        VALUES (user_id, 'Novo Usuário Cadastrado com Sucesso!', NOW());

END //

DELIMITER ;
```

Controle de Acesso de Usuários

```
CREATE USER 'Admin' IDENTIFIED BY 'admin123';
ALTER USER 'Admin' PASSWORD EXPIRE NEVER;
GRANT ALL ON db_teach_up.* TO 'Admin';
FLUSH PRIVILEGES;
```