

SENAC

Curso: Programador WEB

UC 3 - Desenvolver aplicações back-end para web

Professor: Velasco

Aula – Em 28 de outubro de 2022

Visto em sala:

Banco de Dados MySQL - **CRUD**

-- BANCO DE DADOS

-- Acessando o diretório do banco

-- Abrir o prompt(cmd) de comando pelo "executar" digitando CMD

-- CD \

-- CD Xampp\mysql\bin

-- PS: Caso use a console do Xampp é só necessário usar um dos comandos abaixo:

-- mysql -u root -p

-- ou

-- mysql -- user=root -- password=

/*

1 - Fase Conceitual

Rascunho

Ex: Tabelas (Aluno, Curso)

2 - Fase Lógica

Programas de modelagem (BRmodelo, Workbanch, JUDE)

E: Tabela Aluno (id primary key, nome texto, idade int, telefone ...)

3 - Fase Física

Script do Banco

```
CREATE TABLE aluno (  
    idAluno int primary key,  
    nome varchar(100),  
    idade int(3)
```

```
);
```

*/

-- CREATE

-- Cria a Base de dados

CREATE DATABASE IF NOT EXISTS aula_bd;

-- Mostra todos os bancos existentes

SHOW DATABASES;

-- Selecionar o BANCO

USE aula_bd;

-- Criando uma tabela

```
CREATE TABLE aluno(  
    idAluno INT PRIMARY KEY AUTO_INCREMENT,  
    nome VARCHAR(50) NOT NULL,  
    nota01 DOUBLE,  
    nota02 DOUBLE  
);
```

-- Descreve as informações da tabela

DESC aluno;

-- INSERT

-- Inserindo informações na tabela

INSERT INTO aluno (nome, nota01, nota02) VALUES ('Maria', 8., 5);

INSERT INTO aluno VALUES (NULL, 'Paulo', 7.5, 6.7);

INSERT INTO aluno VALUES (NULL, 'João', 5.5, 9.7),

(NULL, 'Julia', 7.5, 5.5) ;

INSERT INTO aluno VALUES (NULL, 'Andre', 5.5, 9.7),

(NULL, 'Amanda', 7.5, 5.5) ;

INSERT INTO aluno VALUES (NULL, 'Julia', 7.5, 5.5);

-- READ - SELECT

-- Exibir todas as informações da tabela

SELECT * FROM aluno;

-- Exibir informações da tabela, selecionando os campos

SELECT idaluno, nome FROM aluno;

-- Exibir as informações da tabela limitando a 2 ocorrências

SELECT * FROM aluno LIMIT 2;

-- Exibir as informações da tabela de somente um usuário

```
SELECT * FROM aluno WHERE idaluno = 1;
```

-- Exibir as informações da tabelas aluno, com a coluna nota01 que tenha valor nulo

```
SELECT * FROM aluno WHERE nota01 IS NULL;
```

-- Exibir as informações da tabela aluno, com a coluna nota01 que NÃO tenha valor nulo

```
SELECT * FROM aluno WHERE nota01 IS NOT NULL;
```

-- Exibir a quantidade de registro de uma Tabela

```
SELECT COUNT(nome) FROM aluno;
```

-- Exibir registros em ordem

```
SELECT * FROM aluno ORDER BY nome;
```

-- Exibir registros em ordem crescente

```
SELECT * FROM aluno ORDER BY nome ASC;
```

-- Exibir registros em ordem decrescente

```
SELECT * FROM aluno ORDER BY nome DESC;
```

-- Exibir registros em ordenado em mais de uma coluna

```
SELECT * FROM aluno ORDER BY nome, nota01;
```

-- Exibir o ultimo registro da tabela

```
SELECT * FROM aluno ORDER BY idaluno DESC LIMIT 1;
```

-- Seleção condicional

-- Exibi nomes que comece com a letra J

```
SELECT * FROM aluno WHERE nome LIKE 'J%';
```

-- Exibi nomes que termina com a letra o

```
SELECT * FROM aluno WHERE nome LIKE '%o';
```

-- Exibir nomes que tenha a letra a ...

```
SELECT * FROM aluno WHERE nome LIKE '%a%';
```

-- Exibir intervalos

```
SELECT * FROM aluno WHERE nota01 BETWEEN 7 AND 8;
```

-- Usamos o operador lógico AND para auxiliar na criação do código de consulta.

-- UPDATE

```
UPDATE aluno SET nota02 = 10 WHERE idaluno = 4;
```

```
UPDATE aluno SET nome = 'Arnaldo', nota01 = 9.9, nota02 = 1.1 WHERE idaluno = 5;
```

-- DELETE

DELETE FROM aluno WHERE idaluno = 4;

-- Excluir todos os registros menos o usuário com id igual a 3:

DELETE FROM aluno WHERE idaluno <> 3;

-- EXCLUIR A BASE

DROP DATABASE IF EXISTS db_aula01;

-- Alterando informações da TABELA

-- Adicionar nova coluna na TABELA

ALTER TABLE aluno ADD media DOUBLE;

-- adicionando uma coluna depois de outra

ALTER TABLE aluno ADD data_nascimento DATE AFTER nome;

-- adicionando uma coluna na primeira posição

ALTER TABLE aluno ADD turma varchar(30) FIRST;

-- Excluindo uma coluna da tabela

ALTER TABLE aluno DROP COLUMN data_nascimento;

-- Alterar o Nome da TABELA

RENAME TABLE aluno TO aluno2;

-- ALTERA NOME DA COLUNA

ALTER TABLE aluno2 CHANGE data_nascimento nascimento date;

-- Alterando o tipo de dados

ALTER TABLE aluno2 MODIFY COLUMN nota02 int;

-- Excluindo todos os dados de uma coluna

UPDATE aluno2 SET nota02 = null;

-- Data em Banco yyyy-MM-DD

SELECT DAY(NOW());

-- EXCLUIR A BASE

DROP TABLE aluno2;

-- RELAÇÃO ENTRE TABELAS

```
CREATE TABLE aluno(  
    idAluno INT PRIMARY KEY AUTO_INCREMENT,  
    nome VARCHAR(50)  
);
```

```

CREATE TABLE disciplina(
    idDisciplina INT PRIMARY KEY AUTO_INCREMENT,
    nomeDisciplina VARCHAR(255) NOT NULL,
    nota01 DOUBLE,
    nota02 DOUBLE,
    id_aluno int,
    FOREIGN KEY(id_aluno) REFERENCES aluno(idAluno)
    ON UPDATE CASCADE ON DELETE CASCADE
);

INSERT INTO aluno VALUES(NULL, 'Jose');
INSERT INTO disciplina VALUES (NULL, 'PHP', 6.3, 8.5, 1);

SELECT * FROM aluno JOIN disciplina ON idaluno = id_aluno;

SELECT nome, nomeDisciplina, nota01, nota02 FROM aluno JOIN disciplina ON idaluno = id_aluno;

```

Não visto em sala. Passados para estudo em casa:

```

/* SELECT 10 + 10
PROJEÇÃO
SELEÇÃO
JUNÇÃO - JUNTAR MAIS DE UMA TABELA
*/
SELECT 10 + 10 AS "SOMA";

-- Excluir somente o aluno João:
DELETE FROM aluno WHERE nome = 'João';

-- Excluir os alunos com id igual a 1 ou 3:
DELETE FROM aluno WHERE idaluno in (1,3);

-- Excluir alunos que estão sem nota01:
DELETE FROM aluno WHERE nota01 = "";

-- Excluir usuários ou com id =2 ou com o nome vitor
DELETE FROM aluno WHERE idaluno =2 OR nome = 'vitor';

```

-- Excluir somente o usuários com nome João e sem nota01:

```
DELETE FROM aluno WHERE nome = 'João' AND nota01="";
```

-- Datas

-- O campo deve estar alimentado no formato YYYY-MM-DD HH:MM:SS para retornar as informações.

-- Para obter o dia atual, faça:

```
SELECT DAY(NOW());
```

-- 2º - Como obter o mês de uma data específica?

```
SELECT MONTH(NOW());
```

-- Para obter o mês atual, faça:

```
SELECT MONTH(NOW());
```

-- 3º - Como obter o ano de uma data específica?

```
SELECT YEAR(DATA) From datas;
```

-- Para obter o ano atual, faça:

```
SELECT YEAR(data) FROM datas;
```

```
INSERT INTO aluno VALUES (null, 'Maria');
```

```
INSERT INTO disciplina VALUES (null, 'Java', 7.2, 1.9, 2);
```

```
INSERT INTO aluno VALUES (null, 'Pedro');
```

```
INSERT INTO disciplina VALUES (null, 'PHP', 2.4, 9.9, 3);
```

```
INSERT INTO aluno VALUES (null, 'Mauro');
```

```
INSERT INTO disciplina VALUES (null, 'PHP', 3.6, 5.9, 4);
```

```
INSERT INTO aluno VALUES (null, 'Carla');
```

```
INSERT INTO disciplina VALUES (null, 'HTML', 4.3, 8., 5);
```

```
INSERT INTO aluno VALUES (null, 'Juliana');
```

```
INSERT INTO disciplina VALUES (null, 'Logica', 9.3, 8.9, 6);
```

```
INSERT INTO aluno VALUES (null, 'Gabriela');
```

```
INSERT INTO disciplina VALUES (null, 'Projetos', 6.5, 9.2, 7);
```

```
INSERT INTO disciplina VALUES (null, 'HTML', 4.3, 8., 1);
```

```
INSERT INTO disciplina VALUES (null, 'HTML', 6.5, 9.2, 3);
```

```
INSERT INTO disciplina VALUES (null, 'HTML', 9.3, 8.9, 4);
```

-- INSERT PARA TESTE

INSERT INTO aluno VALUES (null, null);

INSERT INTO disciplina VALUES (null, 'HTML', 10, 5.2, null);

--RIGHT JOIN

SELECT a.nome, d.nomeDisciplina, d.nota01, d.nota02

FROM aluno a

RIGHT JOIN disciplina d

ON a.idAluno = d.id_aluno;

--LEFT JOIN

SELECT a.nome, d.nomeDisciplina, d.nota01, d.nota02

FROM aluno a

LEFT JOIN disciplina d

ON a.idAluno = d.id_aluno;

SELECT a.nome, d.nomeDisciplina, d.nota01, d.nota02

FROM aluno a

JOIN disciplina d

ON a.idAluno = d.id_aluno;

-- Cria a Base de dados

CREATE DATABASE IF NOT EXISTS aula_bd2;

-- Selecionar o BANCO

USE aula_bd2;

CREATE TABLE ALUNO(

 IDALUNO INT PRIMARY KEY AUTO_INCREMENT,

 NOME VARCHAR(30) NOT NULL,

 SEXO ENUM('M','F') NOT NULL,

 EMAIL VARCHAR(30) UNIQUE

);

CREATE TABLE ENDERECO(

```
    IDENDereco INT PRIMARY KEY AUTO_INCREMENT,  
    RUA VARCHAR(30) NOT NULL,  
    BAIRRO VARCHAR(30) NOT NULL,  
    CIDADE VARCHAR(30) NOT NULL,  
    ID_ALUNO INT UNIQUE,  
    FOREIGN KEY(ID_ALUNO)  
    REFERENCES ALUNO(IDALUNO)  
);
```

```
CREATE TABLE TELEFONE(  
    IDTELEFONE INT PRIMARY KEY AUTO_INCREMENT,  
    TIPO ENUM('CEL','RES','COM') NOT NULL,  
    NUMERO CHAR(10) NOT NULL,  
    ID_ALUNO INT,  
    FOREIGN KEY(ID_ALUNO)  
    REFERENCES ALUNO(IDALUNO)  
);
```

```
-- VERIFICANDO OS NOMES DAS CONSTRAINTS  
-- INFORMATION_SCHEMA = TEM TODAS AS INFORMAÇÕES SOBRE TODAS AS BASES DE DADOS  
USE information_schema;  
  
SELECT * FROM table_constraints WHERE table_schema = 'aula_bd2';
```

```
-- MOSTRA A CONSTRUÇÃO DA TABELA  
SHOW CREATE TABLE ENDereco;
```

```
DESC ALUNO;
```

```
-- EXCLUIR A BASE  
DROP DATABASE IF EXISTS aula_bd2;
```

```
/* CRIANDO AS TABELAS */
```



```
CREATE TABLE ALUNO(  
    IDALUNO INT PRIMARY KEY AUTO_INCREMENT,  
    NOME VARCHAR(30) NOT NULL,  
    EMAIL VARCHAR(30) UNIQUE,  
    SEXO ENUM('M','F') NOT NULL  
);
```

```
CREATE TABLE ENDERECO(  
    IDENDERECO INT PRIMARY KEY AUTO_INCREMENT,  
    RUA VARCHAR(30) NOT NULL,  
    BAIRRO VARCHAR(30) NOT NULL,  
    CIDADE CHAR(2) NOT NULL,  
    ID_ALUNO INT UNIQUE  
);
```

```
ALTER TABLE ENDERECO  
ADD CONSTRAINT FK_ENDERECO_ALUNO  
FOREIGN KEY(ID_ALUNO)  
REFERENCES ALUNO(IDALUNO);
```

```
CREATE TABLE TELEFONE(  
    IDTELEFONE INT PRIMARY KEY AUTO_INCREMENT,  
    TIPO ENUM('RES','CEL','COM') NOT NULL,  
    NUMERO CHAR(8) NOT NULL,  
    ID_ALUNO INT  
);
```

```
ALTER TABLE TELEFONE  
ADD CONSTRAINT FK_TELEFONE_ALUNO  
FOREIGN KEY(ID_ALUNO)  
REFERENCES ALUNO(IDALUNO);
```

```
DESC ALUNO;  
DESC TELEFONE;  
DESC ENDERECO;
```

```
-- VERIFICANDO OS NOMES DAS CONSTRAINTS
-- INFORMATION_SCHEMA = TEM TODAS AS INFORMAÇÕES SOBRE TODAS AS BASES DE DADOS
USE information_schema;
```

```
SELECT * FROM table_constraints WHERE table_schema = 'aula_bd2';
```

```
/*APAGANDO A CONSTRAINT DE CHAVE ESTRANGEIRA*/
```

```
ALTER TABLE TELEFONE
```

```
DROP FOREIGN KEY FK_TELEFONE_ALUNO;
```

```
/* ACOES EM CONSTRAINTS */
```

```
ALTER TABLE TELEFONE
```

```
ADD CONSTRAINT FK_TELEFONE_ALUNO
```

```
FOREIGN KEY(ID_ALUNO)
```

```
REFERENCES ALUNO(IDALUNO)
```

```
ON DELETE CASCADE;
```

```
ALTER TABLE ENDERECO
```

```
DROP FOREIGN KEY FK_ENDERECO_ALUNO;
```

```
ALTER TABLE ENDERECO
```

```
ADD CONSTRAINT FK_ENDERECO_ALUNO
```

```
FOREIGN KEY(ID_ALUNO)
```

```
REFERENCES ALUNO(IDALUNO)
```

```
ON DELETE SET NULL;
```

```
/* ALIMENTANDO AS TABELAS */
```

```
INSERT INTO ALUNO VALUES(NULL,'MARCIO','MARCIO@IG.COM','M');
```

```
INSERT INTO ALUNO VALUES(NULL,'GABRIEL','GABRIEL@IG.COM','M');
```

```
INSERT INTO ALUNO VALUES(NULL,'JOICE','JOICE@IG.COM','F');
```

```
INSERT INTO ALUNO VALUES(NULL,'PRISCILA','PRISCILA@IG.COM','F');
```

```
INSERT INTO ALUNO VALUES(NULL,'JEAN','JEAN@IG.COM','M');
```

```
SELECT * FROM ALUNO;
```

```
INSERT INTO ENDERECO VALUES(NULL,'RUA A','BOTAFOGO','RJ',5);
```

```
INSERT INTO ENDERECO VALUES(NULL,'RUA B','LEME','RJ',6);
```

```
INSERT INTO ENDERECO VALUES(NULL,'RUA C','TIJUCA','RJ',7);
```

```
INSERT INTO ENDERECO VALUES(NULL,'RUA D','FLAMENGO','RJ',8);
```

```
INSERT INTO ENDERECO VALUES(NULL,'RUA E','MORUMBI','SP',9);
```

```
INSERT INTO TELEFONE VALUES(NULL,'CEL','8768799',5);
```

```
INSERT INTO TELEFONE VALUES(NULL,'RES','7874589',5);
```

```
INSERT INTO TELEFONE VALUES(NULL,'COM','4728663',6);
```

```
INSERT INTO TELEFONE VALUES(NULL,'CEL','5256117',6);
```

```
INSERT INTO TELEFONE VALUES(NULL,'RES','6369258',5);
```

```
INSERT INTO TELEFONE VALUES(NULL,'CEL','6539283',8);
```

```
INSERT INTO TELEFONE VALUES(NULL,'COM','9558832',8);
```

```
INSERT INTO TELEFONE VALUES(NULL,'COM','8844583',5);
```

```
INSERT INTO TELEFONE VALUES(NULL,'CEL','8586324',6);
```

```
INSERT INTO TELEFONE VALUES(NULL,'CEL','8745653',6);
```

```
/* ***** SELECAO ***** */
```

```
SELECT NOME FROM ALUNO
```

```
WHERE SEXO = 'F';
```

```
/* JUNCAO */
```

```
SELECT NOME, SEXO, TIPO, NUMERO
```

```
FROM ALUNO,TELEFONE
```

```
WHERE IDALUNO = ID_ALUNO
```

AND

SEXO = 'M';

SELECT NOME, SEXO, BAIRRO, CIDADE

FROM ALUNO,ENDERECO

WHERE IDALUNO = ID_ALUNO;

SELECT NOME, SEXO, TIPO, NUMERO

FROM ALUNO INNER JOIN TELEFONE

ON IDALUNO = ID_ALUNO

WHERE SEXO = 'M';

SELECT NOME, SEXO, TIPO, NUMERO

FROM ALUNO LEFT JOIN TELEFONE

ON IDALUNO = ID_ALUNO;

SELECT NOME,

SEXO,

TIPO,

NUMERO

FROM ALUNO LEFT JOIN TELEFONE

ON IDALUNO = ID_ALUNO;

SELECT NOME,

SEXO,

IFNULL(TIPO,'SEM TIPO') AS "TIPO",

IFNULL(NUMERO,'SEM NUMERO') AS "NUMERO"

FROM ALUNO LEFT JOIN TELEFONE

ON IDALUNO = ID_ALUNO;

SELECT NOME,

```
SEXO,  
  
IFNULL(TIPO,'SEM TIPO') TIPO,  
  
IFNULL(NUMERO,'SEM NUMERO') NUMERO  
  
FROM ALUNO LEFT JOIN TELEFONE  
  
ON IDALUNO = ID_ALUNO;
```

```
SELECT NOME,  
  
SEXO,  
  
IFNULL(TIPO,'SEM TIPO') "TIPO",  
  
IFNULL(NUMERO,'SEM NUMERO') "NUMERO"  
  
FROM ALUNO LEFT JOIN TELEFONE  
  
ON IDALUNO = ID_ALUNO;
```

```
SELECT NOME,  
  
SEXO,  
  
IFNULL(TIPO,'SEM TIPO') 'TIPO',  
  
IFNULL(NUMERO,'SEM NUMERO') 'NUMERO'  
  
FROM ALUNO LEFT JOIN TELEFONE  
  
ON IDALUNO = ID_ALUNO;
```

```
SELECT NOME,  
  
SEXO,  
  
IFNULL(TIPO,'SEM TIPO') AS 'TIPO',  
  
IFNULL(NUMERO,'SEM NUMERO') AS 'NUMERO'  
  
FROM ALUNO LEFT JOIN TELEFONE  
  
ON IDALUNO = ID_ALUNO;
```