

Banco de Dados I

Roteiro de atividades em laboratório

Exercício 01 – Views, Procedures e Funções



UNIFACS
UNIVERSIDADE SALVADOR
LAUREATE INTERNATIONAL UNIVERSITIES

1. Acesse o site <https://www.db4free.net/> e crie um usuário (se necessário).
2. Acesse a opção PhPMysql e entre com o usuário criado.
3. Abra a janela de comandos SQL e execute o trecho destacado em **azul** a seguir.

```
CREATE TABLE tipo ( codtipo char(2) NOT NULL, descricao varchar(20) NULL, PRIMARY KEY (`codtipo`));  
CREATE TABLE apartamento ( numapto int(5) NOT NULL, codtipo char(2) NOT NULL, facilidades  
varchar(100) default NULL, PRIMARY KEY (numapto));  
CREATE TABLE convenio ( codconv int(5) NOT NULL, nome varchar(30) NOT NULL, PRIMARY KEY  
(codconv));  
CREATE TABLE paciente ( codpac int(5) NOT NULL, nome varchar(30) NOT NULL, codconv int(5) NULL, sexo  
char(1) NOT NULL, PRIMARY KEY (`codpac`));  
CREATE TABLE internacao ( codinter int(5) NOT NULL, codpac int(5) NOT NULL, numapto int(5) NOT NULL,  
datainicio datetime NOT NULL, datafim datetime NULL, PRIMARY KEY (codinter));
```

```
INSERT tipo VALUES ('du', 'Duplo'); INSERT tipo VALUES ('lx', 'Luxo'); INSERT tipo VALUES ('si', 'Simples');  
INSERT tipo VALUES ('sl', 'Super Luxo'); INSERT tipo VALUES ('ul', 'Ultra Luxo');
```

```
INSERT apartamento VALUES (100, 'si', NULL); INSERT apartamento VALUES (101, 'si', 'cadeira p/  
acompanhante'); INSERT apartamento VALUES (102, 'si', 'cadeira p/ acompanhante'); INSERT apartamento  
VALUES (103, 'si', NULL); INSERT apartamento VALUES (201, 'du', 'cadeira p/ acompanhante (duas),  
banheiro'); INSERT apartamento VALUES (202, 'du', 'banheiro'); INSERT apartamento VALUES (301, 'lu',  
'cama p/ acompanhante (uma), banheiro'); INSERT apartamento VALUES (302, 'lu', 'cama p/ acompanhante,  
banheiro, tv'); INSERT apartamento VALUES (401, 'sl', 'cama p/ acompanhante, banheiro, tv, frigobar');  
INSERT apartamento VALUES (402, 'sl', 'cama p/ acompanhante, banheiro, tv, frigobar, enfermeiro de  
plantão');
```

```
INSERT convenio VALUES (1, 'Golden Cross'); INSERT convenio VALUES (2, 'Sul America'); INSERT convenio  
VALUES (3, 'PrevMed'); INSERT convenio VALUES (4, 'MediCare'); INSERT convenio VALUES (999,  
'Particular');
```

```
INSERT paciente VALUES (11111, 'John Collins', 999, 'm'); INSERT paciente VALUES (22222, 'Ellen Baker',  
999, 'f'); INSERT paciente VALUES (33333, 'Marcos Pontes', 4, 'm'); INSERT paciente VALUES (44444,  
'Robert Oppenheimer', 1, 'm'); INSERT paciente VALUES (55555, 'Roberta Bondar', 1, 'f'); INSERT paciente  
VALUES (66666, 'Toyohiro Akiyama', 2, 'm'); INSERT paciente VALUES (77777, 'Oleg Atkov', 3, 'm');
```

```
INSERT internacao VALUES (223, 33333, 202, '2007-10-17', '2007-10-01'); INSERT internacao VALUES (445,  
11111, 101, '2007-03-05', NULL); INSERT internacao VALUES (599, 66666, 402, '2007-05-12', '2007-06-12');  
INSERT internacao VALUES (716, 55555, 102, '2007-02-28', NULL); INSERT internacao VALUES (785, 22222,  
401, '2006-10-10', NULL); INSERT internacao VALUES (847, 33333, 103, '2006-06-06', '2006-08-15');
```

Estes comandos vão criar um novo banco de dados com tabelas e dados necessários ao restante do exercício.

Banco de Dados I

Roteiro de atividades em laboratório

Exercício 01 – Views, Procedures e Funções



UNIFACS
UNIVERSIDADE SALVADOR
LAUREATE INTERNATIONAL UNIVERSITIES

4. Baseado nas instruções dadas em aulas anteriores, escreva comandos SQL que atendam aos seguintes requisitos:
 - A) Crie uma **visão** chamada **VW_PACIENTES_INTERNADOS** que exiba o código e o nome de cada paciente internado, o número do apartamento onde ele está e o nome do convênio que o paciente usa.
 - B) Crie uma **stored procedure** chamada **SP_INTERNA_PACIENTE** que recebe como parâmetros de **entrada** um **código de paciente** e um **número de apartamento** e cadastra essa internação com a **data atual** sendo a data de início da internação e a data final em aberto (sem preenchimento). A procedure não deve retornar nenhuma resposta.
 - C) Crie uma **função** chamada **FU_TESTA_PACIENTE** que recebe como **entrada** um código de paciente como parâmetro de entrada e retorna como **saída** o valor 1 (um) se o paciente está internado no momento e 0 (zero) se o paciente não está internado no momento. Lembre-se: os registros de internação cujo campo “**datafim**” está nulo são aqueles que estão internados no momento. Se este campo estiver preenchido significa que a internação já terminou.

RESPOSTAS

- A) Crie uma **visão** chamada **VW_PACIENTES_INTERNADOS** que exiba o código e o nome de cada paciente internado, o número do apartamento onde ele está e o nome do convênio que o paciente usa.

```
create view VW_PACIENTES_INTERNADOS as
select PA.codpac, PA.nome as nome_paciente, IN.numapto, CO.nome as
nome_convênio from paciente PA
inner join internacao IN on PA.codpac = IN.codpac
inner join convenio CO on PA.codconv = CO.codconv;
```

- B) Crie uma **stored procedure** chamada **SP_INTERNA_PACIENTE** que recebe como parâmetros de **entrada** um **código de paciente** e um **número de apartamento** e cadastra essa internação com a **data atual** sendo a data de início da internação e a data final em aberto (sem preenchimento). A procedure não deve retornar nenhuma resposta.

```
create procedure SP_INTERNA_PACIENTE (in parmcod INT, in parmapto INT)
BEGIN
    /* gerando um novo código de internação
    (maior código existente mais um) */
    declare novo int;
    select max(codinter)+1 into novo from internacao;
    /* inserindo nova internação na tabela */
    insert into internacao
        (codinter, codpac, numapto, datainicio, datafim)
        values (novo, parmcod, parmmapto, now(), null);
END;
```

- C) Crie uma **função** chamada **FU_TESTA_PACIENTE** que recebe como **entrada** um código de paciente como parâmetro de entrada e retorna como **saída** o valor 1 (um) se o paciente está internado no momento e 0 (zero) se o paciente não está internado no momento. Lembre-se: os registros de internação cujo campo "**datafim**" está nulo são aqueles que estão internados no momento. Se este campo estiver preenchido significa que a internação já terminou.

```
create function FU_TESTA_PACIENTE (parmcod INT) returns INT
begin
    declare contador int;
    /* verifica se o paciente possui uma internação cadastrada que
    ainda não foi encerrada (ou seja, com datafim preenchido) */
    select count(*) into contador from internacao
    where codpac = parmcod and datafim is null;
    /* se a variável contador for IGUAL a 0 (zero)
    significa que não há internação cadastrada ou que as internações
    do paciente já se encerraram (campo datafim preenchido).
    se a variável contador for DIFERENTE de 0 (zero) significa que já existe
    uma internação em aberto cadastrada para este paciente (o valor do
    contador deve ser 1, pois não faz sentido haver mais de uma internação
    em aberto para o mesmo paciente).
    Logo, não há necessidade de testar o contador.
    Basta retornar o valor do mesmo como resposta */
    return contador;
end
```