**Lista de procedures, fucntions e triggers**

**Athauan Marques da Cunha – 136246-1**

1. Para o banco de dados acadêmico (acadêmico\_proc.sql) criar as seguintes funções:
   1. Criar uma **função** Média Ponderada, para calcular a média do boletim, parâmetros 2 números float, retorno 1 float, cálculo media=(nota1\*0.6)+(nota2\*0.4)

use academico;

drop function IF exists fn\_media;

Delimiter ||

Create function fn\_media (nota1 float, nota2 float)

Returns float

Return ((nota1\*0.6)+(nota2\*0.4));

|| delimiter ;

Select fn\_media(2,5);

* 1. Criar uma tabela ex\_alunos idêntica à estrutura da tabela aluno; Criar uma **trigger** para toda vez que for excluído um aluno da tabela aluno, este aluno ser copiado na tabela ex-aluno

Use acadêmico;

Drop trigger if exists tr\_exAluno;

Create table exAluno like aluno;

Delimiter ||

Creater trigger tr\_exAluno after delete on aluno

For each row

Begin

Insert into exAluno values (old.ra, old.rg, old.nome, old.codcurso);

End;

|| delimiter ;

Delete from aluno where ra=123;

* 1. Faça uma **procedure** para buscar um aluno na tabela aluno por Ra. Se encontrar exibir a msg “existe”, senão exibir “não existe”

Use acadêmico;

Drop procedure IF exists prBuscarAluno;

Delimiter ||

Create procedure prBuscarAluno(in procurador int)

Begin

Declare msg char(30) set msg = “”;

End IF;

End;

|| delimiter ;

* 1. Faça uma **procedure** para alterar a mensalidade de um determinado curso. Passe o código do curso e a taxa de aumento da mensalidade como parâmetros.

use academico;

drop procedure if exists prMensalidade;

delimiter ||

create procedure prMensalidade (in curso char(3), in taxa float)

begin

update curso set mensalidade = mensalidade\*(1+taxa) where codcuro=curso;

end;

||

delimiter ;

call prMensalidade('cc',0.7);

* 1. Crie uma tabela chamada reprovados com todos os alunos que obtiveram médias menor que 6.0 em disciplinas. A tabela deve ter ra aluno, código da disciplina e notaalu. A cópia dos dados dos alunos com média < 6 deve ser feita toda vez que for inserida ou atualizada a média do aluno.

CREATE TABLE reprovados AS SELECT \* FROM boletim WHERE 0;

DESC reprovados;

UPDATE boletim SET notaAlu=10 WHERE ra=231;

DROP TRIGGER reprovar;

DELIMITER $$

CREATE TRIGGER reprovar AFTER UPDATE ON boletim

FOR EACH ROW

BEGIN

IF (new.notaAlu <> old.notaAlu) AND (new.notaAlu<6)THEN

INSERT INTO reprovados VALUES(new.ra, new.codDisc, new.notaAlu);

ELSE

DELETE FROM reprovados WHERE ra=new.ra AND codDisc=new.codDisc;

END IF;

END;

$$

DELIMITER ;

-- UPDATE BOLETIM SET CODDISC='BDD' WHERE RA='765'

UPDATE boletim

SET notaAlu=5.0

WHERE ra=765;

CREATE TABLE reprovados2 AS SELECT \* FROM boletim WHERE 0;

DESC reprovados;

DELIMITER $$

CREATE TRIGGER reprovar2 AFTER INSERT ON boletim

FOR EACH ROW

BEGIN

IF (new.notaAlu<6)THEN

INSERT INTO reprovados VALUES(new.ra, new.codDisc, new.notaAlu);

END IF;

END;

$$

DELIMITER ;

INSERT INTO boletim VALUES ('231','BDOO',3);

* 1. Acrescente uma coluna na tabela cursos para conter o número de alunos em cada curso. Toda vez que um aluno for inserido, incremente este número. Toda vez que uma aluno for deletado, decremente este número.

ALTER TABLE curso ADD nroAlunos INT DEFAULT 0;

SELECT \* FROM curso;

DELIMITER $

CREATE TRIGGER inserir\_alunos AFTER INSERT ON aluno

FOR EACH ROW

BEGIN

UPDATE curso SET nroAlunos=nroAlunos+1 WHERE codCurso = new.codCurso;

END ;

$

DELIMITER ;

-- teste

INSERT INTO aluno VALUES('001','1123','Solemar','AS');

SELECT \* FROM aluno;

UPDATE aluno SET nome='SOLEMAR SILVA' WHERE ra='001';

SELECT \* FROM aluno;

DROP TRIGGER insert\_exaluno;

DELIMITER |

CREATE TRIGGER insert\_exaluno AFTER DELETE ON aluno

FOR EACH ROW

BEGIN

INSERT INTO exalunos VALUES(old.ra, old.rg, old.nome,old.codcurso);

UPDATE curso SET nroAlunos=norAlunos-1 WHERE codCurso=old.codCurso;

END;

|

DELIMITER ;