

Comandos básicos de banco de dados(Sql)

Criar banco de dados

```
create database estudo;
```

Criar tabela

```
create table nome_Tabela
```

Ex:

```
CREATE TABLE nomeDaTabela(  
    cod int not null auto_increment,  
    nome varchar(45),  
    valorDaCompra double,  
    constraint pk_nomeDaTabela primary key(cod)  
);
```

Chave estrangeira:

```
constraint fk_tab1_tab2 foreign key (codigo) references tabela(codigo_);
```

Inserção:

```
insert into consulta values(1,1,1,"2020/01/23",80.00);
```

Atualização:

```
update tabela1 set nome = 'João' where cod = 5;
```

Index único:

```
create UNIQUE INDEX exemplo on peca(p)
```

Consultas:

```
select nome from funcionarios where departamento in ('juridico');  
select departamento from funcionarios group by departamento having avg( salario) > 2000;
```

Controle:

```
create user 'nome do usuario' @ 'localDeOndeTeraAcesso' identified by '0988';  
locais: ip, localhost(próprio Pc), %(qq ip pode acessar)
```

acesso geral ao usuário

grant all on nomedobanco.* to 'nomeDoUsuario'@'localhost';

apenas consultas/leitura:

grant select on nomedobanco.* to 'nomeDoUsuario'@' % ' % ';

Inserção em apenas uma tabela do banco de dados

grant insert on nomedobanco.tabela1 to 'nomeDoUsuario'@' % ' % ';

Excluir acesso do usuário

REVOKE INSERT ON nomedebanco.tabela from 'nomeUsuario' @' % ' % ';

EXCLUIR USUÁRIOS

DROP USER 'nome' @' % ' % ';

Consultar os usuários:

select user from mysql.user;

show grants for 'nomeUsuario' @' % ' % ';

Exibir motores do sql:

Mecanismos de armazenamento que suportam transações:

show engines;

cria uma tabela que suporta transações: antes do ponto e vírgula insere o comando
) engine = InnoDB;

transaction:

start transaction ;

UPDATE contasBancarias SET saldo = saldo -100 where id = 1;

UPDATE contasBancarias SET saldo = saldo + 100 where id = 2;

rollback;

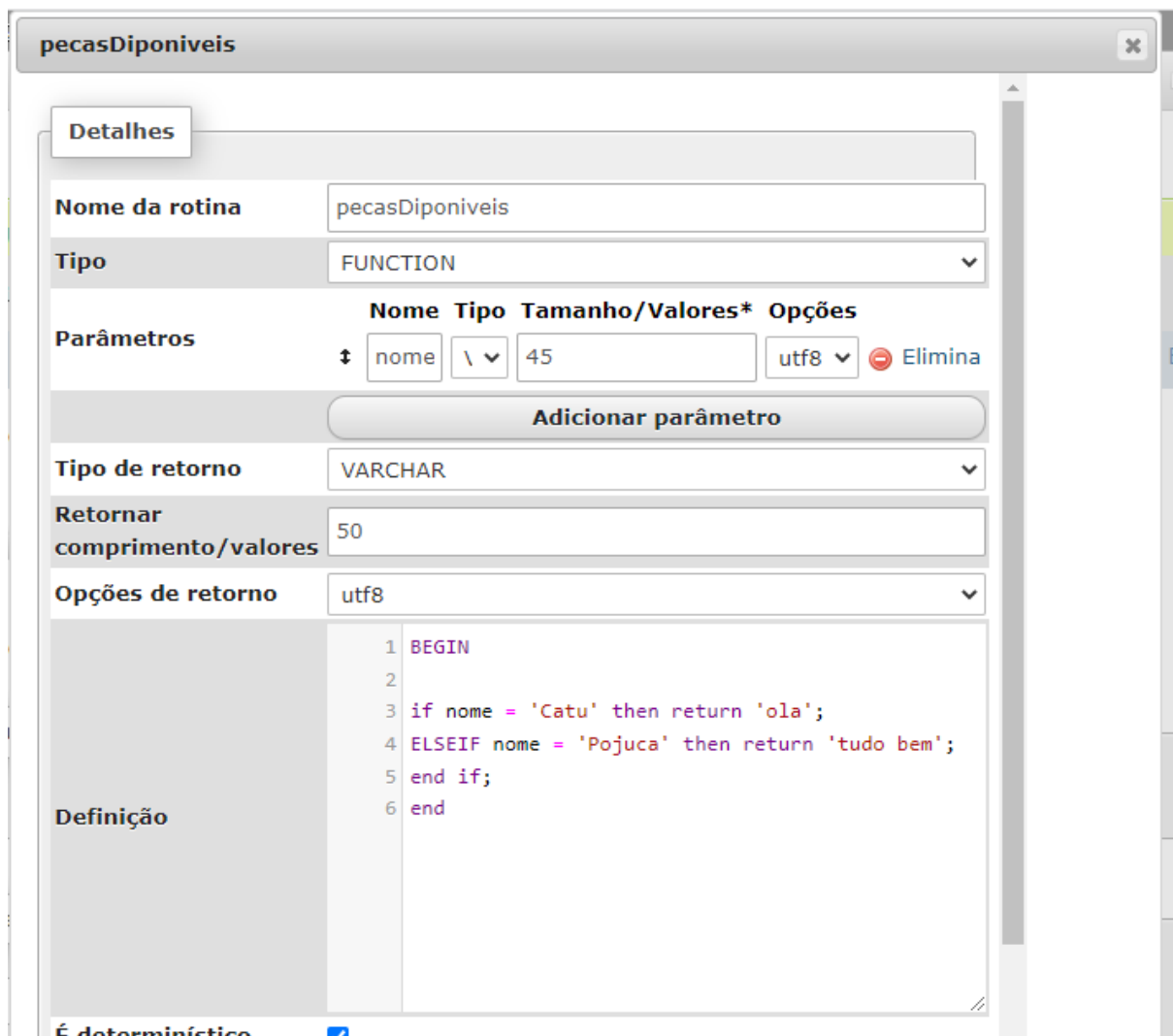
ou commit;

Criar procedimentos/procedures e funções/functions no phpmyadmin

1° Clica no banco de dados desejado

2° Clica em “Rotinas”

3° Clica em adicionar rotina



```
DELIMITER //
```

```
CREATE PROCEDURE nome_procedimento (parâmetros)
```

```
BEGIN /*CORPO DO PROCEDIMENTO*/ END // DELIMITER ;
```

```
CREATE DEFINER='root'@'localhost' PROCEDURE `soma_consulta` (cod int)
BEGIN
```

```
select
nome_medico, sum(valor_consulta) from medico
as m join consulta as c
where m.codigo_medico = c.codigo_medico and cod= c.codigo_medico;
END
call soma_consulta(2);
```

```
CREATE PROCEDURE `atualiza_nota`()

BEGIN

update aula set nota_primeiro_bimestre = 5.0
where nota_primeiro_bimestre = 4.9;
update aula set nota_segundo_bimestre = 5.0
where nota_segundo_bimestre = 4.9;

END
```

Definição	<pre>1 BEGIN 2 3 if nome = 'Catu' then return 'ola'; 4 ELSEIF nome = 'Pojuca' then return 'tudo bem'; 5 end if; 6 end</pre>
É determinístico	<input checked="" type="checkbox"/>
Ajustar privilégios	<input checked="" type="checkbox"/>
Definidor	<input type="text" value="`root`@`localhost`"/>
Tipo de segurança	DEFINER
Acesso de dados SQL	READS SQL DATA
Comentário	

Servidor: 127.0.0.1 » Base de Dados: oficina2 » Tabela: fornecedor

[Procurar](#)
[Estrutura](#)
[SQL](#)
[Pesquisar](#)
[Inserir](#)
[Exportar](#)
[Importar](#)
[Privilegios](#)
[Operações](#)
[Mais](#)

Rastreado
 Accionadores

A mostrar registros de 0 - 4 (5 total, A consulta demorou 0,0030 segundos.)

[SELECT](#) * FROM `fornecedor`

☐ Perfil
 [\[Editar em linha\]](#)
[\[Edita\]](#)
[\[Explicar SQL\]](#)
[\[Criar código PHP\]](#)
[\[Atualizar\]](#)

☐ Mostrar tudo | Número de registros: 25 | Filtrar registros: Ordenar pela chave: Nenhum

+ Opções

	F	fnome	status	cidade
<input type="checkbox"/>	1	Alvaro Santos	20	Catu
<input type="checkbox"/>	2	João Almeida	40	Salvador
<input type="checkbox"/>	3	Júlia Almeida	4	Alagoinhas
<input type="checkbox"/>	4	Carla Alma	3	Catu
<input type="checkbox"/>	5	zé	40	São Paulo

☐ Marcar todos Com os seleccionados:

☐ Mostrar tudo | Número de registros: 25 | Filtrar registros: Ordenar pela chave: Nenhum

Operações resultantes das consultas

```
DROP TRIGGER IF EXISTS baixar_estoque;

DELIMITER |

CREATE TRIGGER baixar_estoque
AFTER INSERT ON vendas
FOR EACH ROW
BEGIN
    UPDATE produto p SET p.quantidade = p.quantidade
    - NEW.quantidade
    WHERE p.codproduto = NEW.codproduto;
END;
```

Definição	1 BEGIN
	2
	3 if nome = 'Catu' then return 'ola';
	4 ELSEIF nome = 'Pojuca' then return 'tudo bem';
	5 end if;
	6 end
É determinístico <input checked="" type="checkbox"/>	
Ajustar privilégios <input checked="" type="checkbox"/>	
Definidor	<input type="text" value="`root`@`localhost`"/>
Tipo de segurança	DEFINER ▼
Acesso de dados SQL	READS SQL DATA ▼
Comentário	<input type="text"/>

Exemplos de função:

```
CREATE DEFINER=`root`@`localhost` FUNCTION
`estoque_baixo`() RETURNS varchar(50) CHARSET
utf8mb4
    READS SQL DATA
    DETERMINISTIC
BEGIN
declare produto_m varchar(30);
select descricao into produto_m from produto
where (quantidade <=5);
return produto_m;
end

select estoque_baixo();
```

```
CREATE FUNCTION `valor_salario` (nome varchar(50))
RETURNS varchar(50)

language sql
reads sql data
deterministic

BEGIN

if nome = 'Técnico' then return 2809;
elseif nome = 'Administrador' then return 3000;
elseif nome = 'Auxiliar financeiro' then return
1967;
elseif nome = 'gestor' then return 7000;
else RETURN 0;
end if;
END
```

#chamar função:

```
select valor_salario('gestor');
```

```
CREATE PROCEDURE `ganha_mais`()
    language sql
    READS SQL DATA
    DETERMINISTIC
BEGIN

select nome from funcao where sal in (select
max(sal) from funcao);
```

```
END  
call ganha_mais;
```

Exportar um banco de dados sql via linha de comando:

```
mysqldump -u root -p bpc > exported_db.sql
```

sem senha

bpc nome do banco

Exportar mais de um banco:

```
mysqldump -u root -p --all-databases > exported_dbs.sql
```

backup via linha de comando: C:\xampp\mysql\bin depois de entrar na pasta digite o comando: `mysqldump -u root -p nome-do-banco > backup.sql`

O download do banco fica dentro da pasta Mysqldump