

TRIGGER e PROCEDURE

Rodar o seguinte script, no MySQL:

```
CREATE TABLE `estado` (  
  `EST_CODIGO` char(2) NOT NULL,  
  `EST_DESCRICAO` varchar(25) DEFAULT NULL,  
  PRIMARY KEY (`EST_CODIGO`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE LOGIN (  
  LOG_TABELA varchar(50) ,  
  LOG_TIPO varchar(50))
```

TRIGGERS

- Trigger de inclusão:

```
DELIMITER $$  
CREATE DEFINER = CURRENT_USER TRIGGER login_incluir BEFORE INSERT ON  
estado  
  FOR EACH ROW BEGIN  
    INSERT INTO login VALUES ('ESTADO','INSERCAO');  
  END$$  
DELIMITER ;
```

- Trigger de alteração:

```
DELIMITER $$  
CREATE DEFINER = CURRENT_USER TRIGGER login_alterar BEFORE UPDATE ON  
estado  
  FOR EACH ROW BEGIN  
    INSERT INTO login VALUES ('ESTADO','ALTERACAO');  
  END$$  
DELIMITER ;
```

- Trigger de exclusão:

```
DELIMITER $$  
CREATE DEFINER = CURRENT_USER TRIGGER login_deletar AFTER DELETE ON  
estado  
  FOR EACH ROW BEGIN  
    INSERT INTO login VALUES ('ESTADO','EXCLUSAO');  
  END$$  
DELIMITER ;
```

Execute o comando à seguir:

```
insert into estado values ('MT','MATO GROÇO')
```

```
select * from LOGIN
```

```
update estado set est_descricao = 'MATO GROSSO' where est_codigo = 'MT'
```

```
select * from LOGIN
```

```
delete from estado where est_codigo = 'MT'
```

```
select * from LOGIN
```

Sendo assim:

*Triggers (ou gatilhos) tem como função disparar após alguma ação. Alterações nas tabelas por meio de operações de inserção, exclusão e atualização de informações ou dados (**insert, delete e update**).*

PROCEDURE

Crie a tabela **cidade** com os seguintes campos, conforme o script:

```
CREATE TABLE `cidade` (  
  `cid_codigo` int(11) NOT NULL,  
  `cid_descricao` varchar(40) DEFAULT NULL,  
  `est_codigo` char(2) NOT NULL,  
  `cod_municipio` int(11) DEFAULT NULL,  
  PRIMARY KEY (`cid_codigo`),  
  KEY `est_codigo` (`est_codigo`),  
  CONSTRAINT `fk_cidade_estado` FOREIGN KEY (`est_codigo`)  
  REFERENCES `estado` (`EST_CODIGO`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

OU

```
CREATE TABLE cidade (  
  cid_codigo INTEGER NOT NULL,  
  cid_descricao VARCHAR(40),  
  est_codigo CHAR(2) NOT NULL,  
  cod_municipio INTEGER);
```

```
ALTER TABLE cidade ADD CONSTRAINT pk_cidade PRIMARY KEY  
(cid_codigo);
```

```
ALTER TABLE cidade ADD CONSTRAINT fk_cidade_estado FOREIGN KEY  
(est_codigo) REFERENCES estado (est_codigo);
```

Assim que criar a tabela, inserir 10 cidades, com estados diferentes correspondentes.

Exemplo:

```
insert into estado values ('GO','GOIAS'),('PI','PIAUI'),('SP','SAO PAULO');
```

```
insert into cidade values
```

```
(1,'GOIANIA','GO',123),(2,'CAMPINAS','SP',234),(3,'TEREZINA','PI',789),(4,'ITUIUTABA','SP',468);
```

- **Procedure de busca direta ou busca com parâmetro**

```
DELIMITER $$
CREATE PROCEDURE `Cidade_Estado` (IN _consulta VARCHAR(40))
BEGIN
    IF(_consulta IS NULL) THEN
        select cid.cid_descricao, est.est_descricao
        from Cidade cid, Estado est
        where cid.est_codigo= est.est_codigo
        order by cid.cid_descricao;
    ELSE
        select cid.cid_descricao, est.est_descricao
        from Cidade cid, Estado est
        where cid.est_codigo= est.est_codigo
        and cid.cid_descricao like CONCAT('%', _consulta ,'%');
    END IF;
END $$
DELIMITER ;
```

- **Rodar no MySQL**

```
call Cidade_Estado(null);
```

ou

```
call Cidade_Estado('N');
```

Definimos então:

Procedures são rotinas criados no BD, identificadas pelo nome de criação, podendo ser executadas. Sendo assim, executam uma série de instruções, recebendo parâmetros e retornando valores.

Exercícios:

1) Crie as tabelas **marca de automóvel** e **automóvel**, com suas principais características. Na sequência, crie triggers de inserção, alteração e exclusão para cada uma das tabelas, para monitorarmos todas as vezes que forem utilizadas estas tabelas, na tabela **login**. Cadastre 11 carros de 4 marcas diferentes.

```
CREATE TABLE LOGIN_MARCA (  
    LOG_TABELA varchar(50) ,  
    LOG_TIPO varchar(50))
```

```
CREATE TABLE LOGIN_AUTOMOVEI (  
    LOG_TABELA varchar(50) ,  
    LOG_TIPO varchar(50))
```

2) Crie stored procedure de busca com o nome do automóvel, sua placa, RENAVAM, chassi, capacidade de pessoas e sua marca.

3) Crie as tabelas **aluno**, **turma** e **curso**, com suas principais características. Agora, crie triggers de inserção, alteração e exclusão para cada uma das tabelas, para monitorarmos todas as vezes que forem utilizadas estas tabelas, na tabela **login**. Inserir informações de 7 alunos, 5 turmas e 2 cursos.

4) Crie stored procedure de busca com o nome do aluno, sua matrícula, ID, CPF, endereço, CEP, sua **turma** e qual o **curso** que está fazendo.