

```
use rede_restaurantes;

drop table if exists log_pedidos;
```

```
-- Criar tabela de log para o TRIGGER 1 e 2 da tabela 1
```

```
CREATE TABLE log_pedidos (
  id_pedido INT NOT NULL AUTO_INCREMENT,
  id_loja INT NOT NULL,
  valor_pedido DECIMAL(5,2) NOT NULL,
  data_pedido DATETIME NOT NULL,
  forma_pagamento varchar(30) NOT NULL DEFAULT 'Cartão de Crédito',
  nome_cliente VARCHAR(30),
  cpf_cliente VARCHAR(20),
  id_colaborador INT,
  acao_registro VARCHAR (20),
  PRIMARY KEY (id_pedido),
  FOREIGN KEY (id_loja) REFERENCES lojas_franquia (id_loja));
```

```
-- Criar o TRIGGER 1/ TABELA 1
```

```
-- Inserir na tabela de log os valores antigos (antes do update)
```

```
DROP TRIGGER if exists update_pedidos;
```

```
CREATE TRIGGER `update_pedidos`
BEFORE UPDATE ON `pedidos`
FOR EACH ROW
INSERT INTO `log_pedidos` (id_pedido, id_loja, valor_pedido, data_pedido, forma_pagamento,
nome_cliente, cpf_cliente, id_colaborador, acao_registro )
VALUES (OLD.id_pedido, OLD.id_loja, OLD.valor_pedido, OLD.data_pedido,
OLD.forma_pagamento,
OLD.nome_cliente, OLD.cpf_cliente, OLD.id_colaborador, 'update' );
```

-- Trigger 2/ TABELA 1: depois de deletar um registro na tabela pedidos, inserir valores no log

```
CREATE TRIGGER `delete_pedidos`  
AFTER DELETE ON `pedidos`  
FOR EACH ROW  
INSERT INTO `log_pedidos` (id_pedido, id_loja, valor_pedido, data_pedido, forma_pagamento,  
nome_cliente, cpf_cliente, id_colaborador, acao_registro )  
VALUES (OLD.id_pedido, OLD.id_loja, OLD.valor_pedido, OLD.data_pedido,  
OLD.forma_pagamento,  
OLD.nome_cliente, OLD.cpf_cliente, OLD.id_colaborador, 'delete' );
```

-- Criação da tabela para TRIGGER 1/ TABELA 2

-- TRIGGER para fazer um log de cnpps toda vez que houver um update

```
use rede_restaurantes;  
drop table if exists log_cnpj;  
-- Criar tabela de log
```

```
CREATE TABLE log_cnpj (  
cnpj_f VARCHAR(20),  
id_item INT NOT NULL,  
nome_item VARCHAR(50) NOT NULL DEFAULT 'Combo Supreme',  
id_loja INT NOT NULL,  
acao_registro VARCHAR(20),  
PRIMARY KEY (cnpj_f),  
FOREIGN KEY (id_loja) REFERENCES lojas_franquia (id_loja) );
```

-- Criar o TRIGGER 1 / TABELA 2

-- Inserir na tabela de log os valores antigos (antes do update)

```
DROP TRIGGER IF EXISTS update_cnpj;
```

```
CREATE TRIGGER `update_cnpj`  
BEFORE UPDATE ON `fornecedores`  
FOR EACH ROW  
INSERT INTO `log_cnpj` (cnpj_f, id_item, nome_item, id_loja, acao_registro )  
VALUES (cnpj_f, id_item, nome_item, id_loja, 'update' );
```

-- Trigger 2 / TABELA 2: depois de deletar registro nos fornecedores, adicionar no log

```
CREATE TRIGGER `delete_cnpj`  
AFTER DELETE ON `fornecedores`  
FOR EACH ROW  
INSERT INTO `log_cnpj` (cnpj_f, id_item, nome_item, id_loja, acao_registro )  
VALUES (OLD.cnpj_f, OLD.id_item, OLD.nome_item, OLD.id_loja, 'delete' );
```