

Os pedidos de refeição em uma rede de restaurantes são armazenados em um banco de dados cuja especificação (em PostgreSQL) encontra-se a seguir. Os preços dos pratos podem variar de uma loja para outra e nem todos os restaurantes oferecem os mesmos pratos. Além disso, cada pedido deve ser feito a somente um único restaurante. Escreva, utilizando triggers PostgreSQL, a implementação da restrição de integridade que garante a consistência dos pedidos no que se refere somente a unicidade de restaurante do pedido.

```
CREATE TABLE restaurante (  
    cnpj integer NOT NULL,  
    endereco character varying NOT NULL,  
    CONSTRAINT rest_pk PRIMARY KEY (cnpj));  
  
CREATE TABLE prato (  
    prato_id integer NOT NULL,  
    nome character varying NOT NULL,  
    CONSTRAINT prato_pk PRIMARY KEY  
    (prato_id));  
  
CREATE TABLE menu (  
    cnpj integer NOT NULL,  
    prato_id integer NOT NULL,  
    preco real NOT NULL,  
    CONSTRAINT menu_pk PRIMARY KEY  
    (cnpj,prato_id),  
    CONSTRAINT menu_rest_fk FOREIGN KEY  
    (cnpj) REFERENCES restaurante (cnpj),  
    CONSTRAINT menu_prato_fk FOREIGN KEY  
    (prato_id) REFERENCES prato (prato_id));
```

```
CREATE TABLE pedido (  
    pedido_id integer NOT NULL,  
    cnpj integer NOT NULL,  
    CONSTRAINT pedido_pk PRIMARY KEY  
    (pedido_id),  
    CONSTRAINT pedido_rest_fk FOREIGN KEY  
    (cnpj) REFERENCES restaurante (cnpj));  
  
CREATE TABLE item_pedido (  
    pedido_id integer NOT NULL,  
    item integer NOT NULL,  
    cnpj integer NOT NULL,  
    prato_id integer NOT NULL,  
    qtd integer NOT NULL,  
    CONSTRAINT item_pk PRIMARY KEY  
    (pedido_id,item),  
    CONSTRAINT item_pedido_fk FOREIGN KEY  
    (pedido_id) REFERENCES pedido  
    (pedido_id),  
    CONSTRAINT item_menu_fk FOREIGN KEY  
    (cnpj,prato_id) REFERENCES menu  
    (cnpj,prato_id));
```