

Exercicio Pratico DB

Para criar uma (simples) base de dados usando grafos, são precisos apenas dois passos:

- Definir a table de nodos
- Definir a tabela das arestas

Exemplo de uma database de grafos

Como exemplo, vamos considerar uma base de dados de gestão de entrega e pedidos de comida por parte de restaurantes

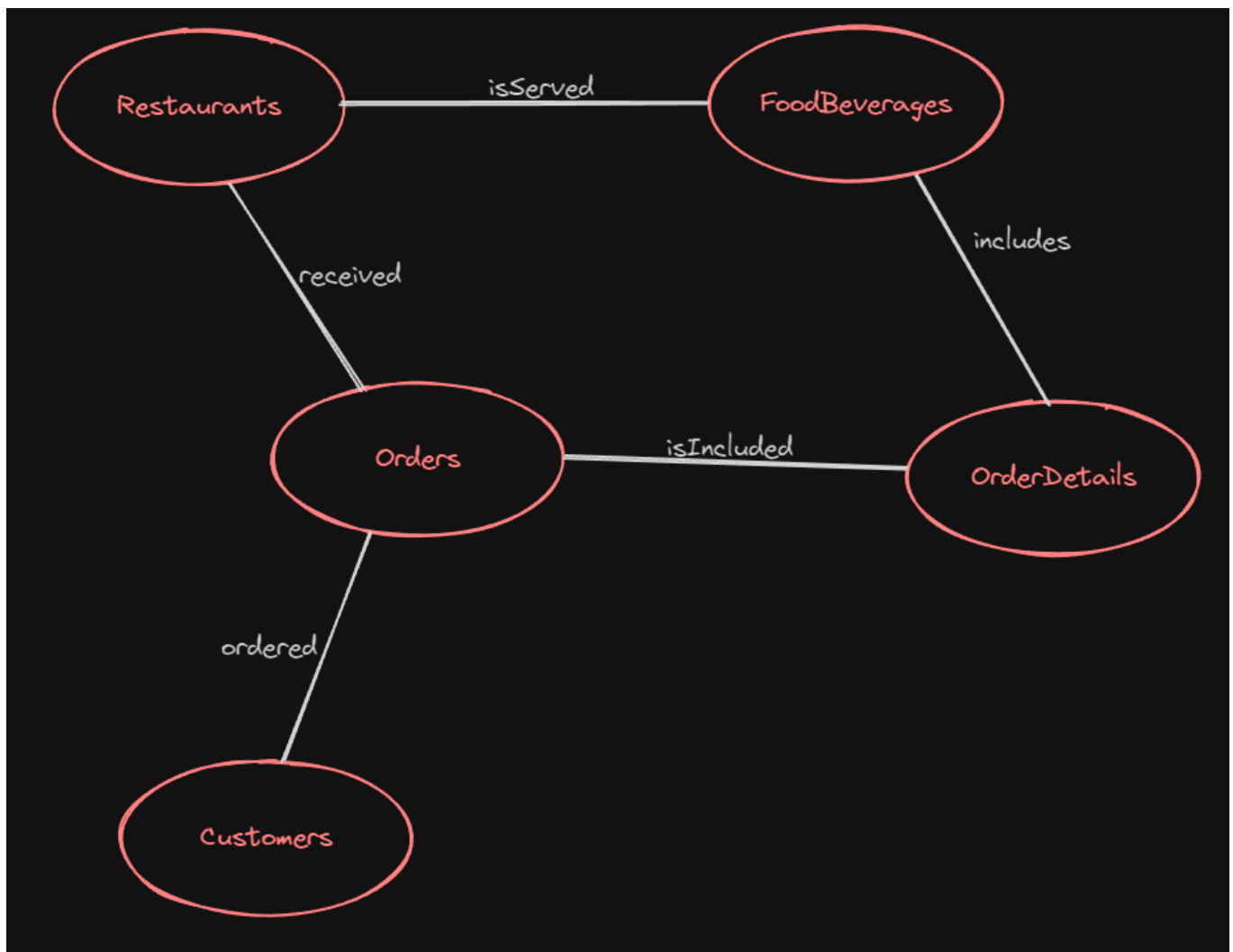
Para este exemplo vamos apenas considerar que temos um restaurante, e vários pedidos.

Para os nodos, temos:

1. *Restaurants*: Dados sobre os diferentes restaurantes.
2. *Locations*: Localização dos restaurantes e dos clientes
3. *Orders*: Dados sobre os pedidos feitos pelos clientes
4. *OrderDetails*: Detalhes das ordens
5. *FoodBevarages*: Dados da comida e bebidas
6. *Customers*: Dados dos consumidores

Para as arestas, temos:

1. *isServed*: Se a comida é servida no restaurante ou não.
2. *isLocated*: Se o restaurante é localizado
3. *willReceiveIn*: Se o cliente vai receber o pedido na localização
4. *ordered*: Se o cliente fez o pedido
5. *isIncluded*: Se a comida está incluída no pedido
6. *received*: Se o pedido foi recebido pelo restaurante
7. *include*: Se o pedido inclui Comida ou Bebida



Criar database grafos

```
CREATE TABLE Restaurants
(
  [RestaurantID] INT PRIMARY KEY NOT NULL,
  [Name] VARCHAR(20) NOT NULL,
  [Description] VARCHAR(50) NOT NULL,
  [Opening] TIME NOT NULL,
  [Closing] TIME NOT NULL,
  [DeliveryCharge] SMALLMONEY DEFAULT 0 NOT NULL
) AS NODE
```

```
CREATE TABLE FoodBeverages
(
  [FoodBeverageID] INT PRIMARY KEY NOT NULL,
  [Name] VARCHAR(50) NOT NULL,
```

```
[Amount] MONEY NOT NULL
```

```
) AS NODE
```

```
CREATE TABLE Orders
```

```
(
```

```
[OrderID] INT PRIMARY KEY NOT NULL,
```

```
[CustomerID] INT NOT NULL,
```

```
[OrderDate] DATETIME NOT NULL,
```

```
[TotalCost] MONEY NOT NULL,
```

```
[RestaurantID] INT NOT NULL
```

```
) AS NODE
```

```
CREATE TABLE OrderDetails
```

```
(
```

```
[OrderDetailID] INT PRIMARY KEY NOT NULL,
```

```
[OrderID] INT NOT NULL,
```

```
[FoodBeverageID] INT NOT NULL,
```

```
[Quantity] TINYINT NOT NULL,
```

```
[UnitCost] MONEY NOT NULL,
```

```
[ComputedCost] MONEY NOT NULL
```

```
) AS NODE
```

```
CREATE TABLE Customers
```

```
(
```

```
[CustomerID] INT PRIMARY KEY NOT NULL,
```

```
[Name] VARCHAR(50) NOT NULL,
```

```
) AS NODE
```

```
CREATE TABLE isServed AS EDGE
```

```
CREATE TABLE ordered AS EDGE
```

```
CREATE TABLE isIncluded AS EDGE
```

```
CREATE TABLE received AS EDGE
```

```
CREATE TABLE includes AS EDGE
```

```
CREATE UNIQUE INDEX ix_isIncludes_from_to on isIncluded ($from_id, $to_id)
```

```
CREATE UNIQUE INDEX ix_includes_from_to on includes ($from_id, $to_id)
```

```
CREATE UNIQUE INDEX ix_ordered_from_to on ordered ($from_id, $to_id)
```

```
CREATE UNIQUE INDEX ix_received_from_to on received ($from_id, $to_id)
```

```
CREATE UNIQUE INDEX ix_isServed_from_to on isServed ($from_id, $to_id)
```

Inserir dados para teste

Primeiramente inserimos dados na tabela de nodos, de forma semelhante ao processo de uma base de dados relacional

```
INSERT INTO Restaurants
VALUES
(1, 'Subway','American * Sandwiches * Fast Food * Salad','10:00','15:59',0)

INSERT INTO FoodBeverages
VALUES
(1,'Sliced Chicken and Ham',149.50),
(2,'Subway Melt',184.00),
(3,'Meatball Marinara Regular Meal',281.75),
(4,'Coke in Can',51.75)
```

A seguir temos que realizar as "conexões". É importante referir que para tal, precisamos dos ID's de ambos os nodos a serem ligados. Felizmente, estes são gerados automaticamente pelo servidor, e podem ser obtidos facilmente:

```
-- food&beverages served on Subway (RestaurantID = 1)
INSERT INTO isServed
VALUES
-- for the Sliced Chicken and Ham
(
  (SELECT $node_id FROM Restaurants r WHERE r.RestaurantID = 1),
  (SELECT $node_id FROM FoodBeverages fb WHERE fb.FoodBeverageID = 1)
),
-- for the Subway Melt
(
  (SELECT $node_id FROM Restaurants r WHERE r.RestaurantID = 1),
  (SELECT $node_id FROM FoodBeverages fb WHERE fb.FoodBeverageID = 2)
),
-- for the Meatball Marinara Regular Meal
(
  (SELECT $node_id FROM Restaurants r WHERE r.RestaurantID = 1),
  (SELECT $node_id FROM FoodBeverages fb WHERE fb.FoodBeverageID = 3)
),
-- for the coke in can
(
```

```
(SELECT $node_id FROM Restaurants r WHERE r.RestaurantID = 1),  
(SELECT $node_id FROM FoodBeverages fb WHERE fb.FoodBeverageID = 4)  
)
```

Exemplo na visualização

A seguir vamos explorar um exemplo de uma query.

O objetivo desta query vai ser de listar o nome dos restaurantes e da comida/bebida, bem como o preço total, nos casos em que o pedido foi servido.

```
SELECT Restaurants.Name, FoodBeverages.Name, FoodBeverages.Amount  
FROM Restaurants, isServed, FoodBeverages  
WHERE MATCH(FoodBeverages<-(isServed)-Restaurants)
```

As primeiras duas linhas são familiares no esquema relacional, porém a terceira merece ser analisada.

O que vai fazer é nada mais do que indicar o nodo1, aresta e o nodo2. A seta indica qual o nodo1 e o nodo2, isto é, qual o id do nodo1 e o id do nodo2.

Exercício

De forma a imitar o sistema de "clientes compraram isto com...", pretende-mos, através de uma *query*, ver quais outros produtos foram comprados juntamente com um certo produto.

Por exemplo, supomos que o cliente 1 realizou uma ordem com os produtos foodBeverageID 1 e 2 (Sliced Chicken and Ham e Subway Melt). Ao realizar a pesquisa sobre o produto 1, pretende-se obter o nome do produto 2 (Subway Melt).

Passos:

1. Inserir um registo na tabela *Orders*.
2. Inserir os registos na tabela *OrderDetails*.
3. Inserir as "conexões" nas tabelas *includes* e *isIncluded*.
4. Realizar a pesquisa.

A pesquisa pelo grafo vai ser:

