relational database designed according to the definitions in Tables 1-6 above to define lists of customers, addresses, product types, baskets, positions of products in a basket for each order, and orders. In Tables 7-12, key field headings are denoted with an asterisk. Customers Table 7 defines customers and each customer is uniquely identified by a CustomerId field. Customers Table 7 also includes a NAME field and a foreign key field AddressId that links addresses in an Addresses table to customers.

TABLE 7

Customers

CustomerId*	NAME	AddressId
1	John Smith	1
2	David Klein	2

[0103] Addresses Table 8 defines addresses having a town and a street. The Address id itself is a valid unique key for an address and the connection between address and customer is done through the Customers Table 7 (AddressID field).

TABLE 8

Addresses

AddressId*	Town	Street
1	Athens	Main Street
2	Louisville	Willow Avenue

[0104] Table 9 defines products having names with key ProductId.

TABLE 9

Products

ProductId*	Name
1	Saw
2	Hammer
3	Wrench
4	Screwdriver

[0105] Table 10 defines shopping baskets having customers with key BasketId.

TABLE 10

Baskets

BasketId*	CustomerId
1	2
2	1

[0106] Table 11 defines positions of orders in baskets and having products. Positions are dependent on the existence of baskets and orders so the primary key for positions is a combination of PositionId, BasketId, and OrderId.

TABLE 11

Positions

PositionId*	BasketId*	OrderId*	ProductId
1	1	3	2
2	1	2	3
3	2	1	4

[0107] Table 12 defines orders having customers and indicating whether or not each order is submitted with primary key OrderId.

TABLE 12

Orders

OrderId*	CustomerId	Submitted
1	1	False
2	2	False
3	2	False

[0108] As shown in **FIG. 5**, process **150** defines the database operations on backend database **22** that are needed for this simple task using these tables 7-12. Process **150** includes front end application program **12** receiving **(152)** a name of a customer. Process **150** includes the business software application querying **(154)** a database with Customers table (Table 7) for the name in the NAME field. Process **150** includes checking if the customer's name matches **(156)** a row in the Customers table (Table 7). If no match is made, process **150** includes the business software application obtaining **(158)** the address of the customer, inserting **(160)** a new row in the Address table (Table 8) with a new AddressID and address, and inserting **(162)** a new row in the Customers table (Table 7) with a new CustomerId and the AddressID. If a match is made, process **150** includes the business software obtaining **(164)** a name of a product to order for the customer. Process **150** includes the business software querying **(166)** the Products table (Table 9) for the product name.

[0109] Process **150** includes checking if the product name matches **(168)** a row in the Products table (Table 9). If a match is made, then process **150** includes the business software inserting **(170)** a new order in the Orders table (Table 12) with the customer's CustomerId and setting the Submitted field to "False". Otherwise, process **150** returns to obtaining **(164)** the name of the product to order. Process **150** includes the business software inserting **(172)** a new basket in the Basket table (Table 10) with the customer's CustomerId.

[0110] Process **150** includes the business software inserting **(174)** a new position in the Positions table (Table 11) with the CustomerId, BasketId, and ProductId. Process **150** includes the business software receiving (176) a request to submit the order. Process **150** includes the business software querying **(178)** the Orders table (Table 12) by the customer's CustomerId and this query returns orders matching the customer's CustomerId. Process **150** includes the business software selecting **(180)** orders in the Orders table (Table 12) matching the orders for the customer's CustomerId. Process **150** includes the business software setting **(182)** the Sub-