Database Design and Report Package

Jason Goncalves

IT-235

Southern New Hampshire University

**Purpose, Goals, and Objectives**

Dylan and Samantha Morgan of Cougar Pizza Pies, requests that their database will allow them to track customer orders. These orders will consist of pizza type (thin pizza, regular pizza, or pan pizza), limited toppings (pepperoni, sausage, ham, and extra cheese), total pizza price, and notes for custom customer instruction. The database needs to report and track customer orders by street address, city, state, zip code, and phone number. The Morgan’s will use this information for marketing and research purposes. The goal is to help ensure that Cougar Pizza Pies is profitable and successful. The Morgan’s plan to open a chain at other college towns if this goal is accomplished.

**User Requirements**

This database must collect and report pizza orders which include (This database must be able to handle multiple stores): Added toppings and pizza type (Customers can order more than one pizza, amount of pizza for the order will be collected), the customer who ordered the pizza, the customer’s delivery address (Customer details may change, customers are preferred to have one address, can change if needed), name and phone number of the employee delivering the pizza, the name of the employee who took the pizza order ,Sales per hour, day, week, and month, and the delivery status detail at various times of the night. Each pizza order should have a status, an order date, order time, delivery date, delivery time and an optional note that states if the order will be picked up by the customer (if the order is picked up by the customer, the “pickup time” is the “delivery time”).

**Business Process and Restriction**

The database needs to report multiple pizza orders which include pizza type and toppings, the customer who ordered the pizza, the address that the pizza was delivered to, the name and phone number of the delivery employee, name of the employee who took the pizza order, sales per hour, per day, per week, and per month, delivery status details at various points of the sales night. Each order should have a status along with the order date and order time along with delivery date and delivery time. Also, there may be a note stating if the customer will pick up the order, which if that is so, the delivery time will be the customer pick up time. A customer can also order more than one pizza, the customer may change their preferred address and may order from a different address; the database must track all addresses entered for the customer but keep one address as the customer default address. Finally, the database will keep track of the placed order time and delivery time to ensure that the orders are delivered within the hour.

**End Result**

The database's desired result will need to be user-friendly, secure, and allow for expansion for new store locations. The data will need to be stored correctly and pull and display the order status to customers, employees, or owners. With all the information we have collected, the Morgans can track sales per hour, day, week, and month. The Morgans will be able to populate business metrics to observe trends and see where transactions are lacking; thus, allowing the Morgans to expand more efficiently when needed.

.

**II. Conceptual Design**

Cougar Pizza Pies is looking to accomplish a specific type of business model and database to have a successful business. There are a few different options for ERMs that we could execute, the most common being the Chen Notation, the Crow's Foot Notation, and the UML Notation. The Chen Notation and the UML notation use conceptual modeling. The only difference is that the UML notation applies some implementation modeling; this allows for some flexibility for the database. The primary purpose of the Crow's Foot Notation is implementation-oriented models. Some information we acknowledge for entities and attributes is customer information, which includes name, address, and phone number. We will also gather data regarding the order, such as type, toppings, and delivery data.

The owners of Cougar Pizza Pies (Dylan and Samantha Morgan) have a goal in mind, and that's to validate their business model at their primary location to then branch off to other store locations. The best ERM to quickly roll out and create a compelling and straightforward database would be the Crow's Foot Notation. The creation of this ERM will allow us to link the entities to one another while also identifying the relationships between one another by utilizing Primary Keys (PK) and Secondary Keys (FK).

**Entities**

**CUSTOMER = RED, CustomerID (PK), PreferredAddressID (FK), CustomerNote, CustomerLastName, CustomerFirstName, CustomerPhone, CustomerEmail**

**ADDRESS = Green, CustomerAddressID (PK), StreetLine1, StreetLine2Apt, City, State, ZipCode**

**EMPLOYEE = Blue, EmployeeID (PK), EmployeeLastName, EmployeeFirstName, EmployeePhone**

**ORDER = PINK, OrderID (PK), CustomerID (FK), AddressID (FK), PlacedByEmployeeID (FK), DeliveryEmployeeID (FK), DateTimeOrdered, PickupOrDelivery, DateTimeDelivered, StatusID (FK), StoreID, SpecialDetails**

**ORDER DETAIL = Purple, OrderID (PK) (FK), PizzaOrderID (PK), PizzaTypeID (FK), PizzaPrice**

**TOPPING ORDER DETAIL = Orange, OrderID (PK) (FK), PizzaOrderID (PK) (FK), ToppingOrderID (PK), ToppingID (FK), ToppingPrice**

**PIZZA TYPE = Yellow, PizzaTypeID (PK), PizzaPrice, PizzaTypeDetails**

**TOPPING = Light Purple, ToppingID (PK), ToppingDesc, ToppingPrice**

**DELIVERY STATUS = Gray, StatusID (PK), StatusDesc**

**STORE = Turqoise, StoreID (PK), StoreDesc**

**Entities and Attributes**

|  |  |
| --- | --- |
| **CUSTOMER** | |
| **CustomerID (PK)** | **ID for the customer** |
| **PreferredAddressID (FK)** | **Address ID from address** |
| **CustomerNote** | **Notes for customer** |
| **CustomerLastName** | **Customers last name** |
| **CustomerFirstName** | **Customers first name** |
| **CustomerPhone** | **Customers phone number** |
| **CustomerEmail** | **Customers email address** |

|  |  |
| --- | --- |
| **ADDRESS** | |
| **CustomerAddressID (PK)** | **ID for customer address** |
| **StreetLine1** | **Customer’s street** |
| **StreetLine2Apt** | **Customer’s street** |
| **City** | **Customer’s city** |
| **State** | **Customer’s State** |
| **ZipCode** | **Customer’s Zip Code** |

|  |  |
| --- | --- |
| **EMPLOYEE** | |
| **EmployeeID (PK)** | **ID to identify employee** |
| **EmployeeLastName** | **Employee’s last name** |
| **EmployeeFirstName** | **Employee’s first name** |
| **EmployeePhone** | **Employee’s phone number** |

|  |  |
| --- | --- |
| **ORDER** | |
| **OrderID (PK)** | **ID identifying each order** |
| **CustomerID (FK)** | **Customer’s ID** |
| **AddressID(FK)** | **Customer address ID** |
| **PlacedByEmployeeID (FK)** | **Employee ID that takes order** |
| **DeliveryEmployeeID (FK)** | **Employee ID that delivers order** |
| **DateTimeOrdered** | **Date and time order placed** |
| **PickupORDelivery** | **Picking up or delivery pizza** |
| **DateTimeDelivered** | **Date and time pizza delivered** |
| **StatusID(FK)** | **Status of order** |
| **StoreID** | **ID of store selling the pizza** |
| **SpecialDetails** | **Special order details** |

|  |  |
| --- | --- |
| **ORDER DETAIL** | |
| **OrderID (PK) (FK)** | **ID for the order** |
| **PizzaOrderID (PK)** | **Pizza ID to identify the pizza** |
| **PizzaTypeID (FK)** | **ID for the pizza type** |
| **PizzaPrice** | **Price of the pizza** |

|  |  |
| --- | --- |
| **TOPPING ORDER DETAIL** | |
| **OrderID (PK) (FK)** | **Id to identify order** |
| **PizzaOrderID (PK)(FK)** | **Id for pizza order** |
| **ToppingOrderID (PK)** | **Identifies topping order** |
| **ToppingID (FK)** | **Identifies the topping** |
| **ToppingPrice** | **Price of the topping** |

|  |  |
| --- | --- |
| **PIZZA TYPE** | |
| **PizzaTypeID (PK)** | **Pizza ID to identify pizza type** |
| **PizzaPrice** | **Price for that type of pizza** |
| **PizzaTypeDetails** | **Details for the pizza type** |

|  |  |
| --- | --- |
| **TOPPING** | |
| **ToppingID (PK)** | **Id for the topping** |
| **ToppingDesc** | **Topping description** |
| **ToppingPrice** | **Price for topping** |

|  |  |
| --- | --- |
| **DELIVERY STATUS** | |
| **StatusID (PK)** | **Id for status** |
| **StatusDesc** | **Status description** |

|  |  |
| --- | --- |
| **STORE** | |
| **StoreID (PK)** | **Id for the store** |
| **StoreDesc** | **Store description** |

**DESIGN REVISIONS**

**ERD 1:**

A screenshot of a computer

Description automatically generated

**Dependency Chart:**

A screenshot of a social media post

Description automatically generated

**FINALIZED DESIGN**

**ERD 2:**

A screenshot of a computer

Description automatically generated

**SUMMARY**

With this final design, Cougar Pizza Pies will be able to track their orders more efficiently. The database we'll implement has ten entities related to one too many relationships and more. If the Morgans need to expand the database they can, the new database is designed for future expansion. For the normalization of the database, we took each entity's attributes through the stages of normalization until we reached the final 3NF dependency. End-user will not be able to input incorrect data due to the established relationships and restrictions. For the Morgans, this database will achieve the desired result with the possibility of expansion.