Lunds Universitet **11 april 2016**
Institution för Datavetenskap

Projekt - KrustyCoockies
EDA216 - Databasteknik

Christine Boghammar, dat12cbo@student.lu.se
Erik Söderberg, dat12eso@student.lu.se

Innehåll

1 Introduction 1

2 Requirements 1
2.1 Production 1
2.2 Raw Materials 1
2.3 Recipe 1
2.4 Produced Pallets 2
2.5 Orders and Production 2
2.6 Delivery 2

3 System outline 2
3.1 Database Manager 2
3.2 Language 2

4 ER - Diagram 3

5 Relations 3

6 SQL Statements 3

7 User Manual 4

1 Introduction

The purpose of the project is to get a deeper knowledge of how a database is implemented och operates with a client as it is represented with an interface. The objective was to create a program for Krusty Cookies which would handle the production of cookies. From the given information a program was constructed using a interface based on Java Swing and a database that handled the production.

2 Requirements

With the implementation of Krusty Cookies some requirements had to be fulfilled in order for the program to operate with its purpose.

2.1 Production

All the information about costumers, cookies, recipes and ingredients are stored in the database. With that information a pallet of cookies can be produced and stored in the database with its unique pallet number. One order can have many pallets assigned to it and one pallet can only be assigned to one order. One order must have been placed before one pallet can be produced. The purpose is to later know when a order has been completely handled and can have its pallets delivered. This is not yet implemented as it was not a requirement for this part of the system. With this being said, this program only handles production, blocking and searching of pallets. The program will only handle smart users as it is only a prototype and not able to handle bad inputs.

2.2 Raw Materials

Raw materials are, as of now, stacked up with an initial amount of 10 kg. The requirements with this program was to control that there was enough raw materials in stock when a pallet of cookies were to be produced. Otherwise the production of the cookie pallet should not be possible. With further implementations, refill of raw materials could easily be handled but is not required with this part of the program. It is not possible to visually see the stock amount of raw materials in the interface with this version of the program.

2.3 Recipe

Recipes are introduced into the database from which the the amount needed for production of pallets are collected. It is not possible to modify any recipes with the interface.

2.4 Produced Pallets

Any information for production of pallets are accessible in the interface. Pallets of cookies can be produced if there are enough ingredient in store and a produced pallet is displayed in the interface with the name of the cookie and date of production. Several search functions are available and the possibility of blocking a certain cookie that have not been delivered (located in the freezer).

2.5 Orders and Production

Orders are not handled as of this point since it was not included in the requirements for this program. But the implementation for orders are in some length handled to be further implemented in the future. As of now, produced pallets are connected to a certain order. The way the program will handle orders is to show the created orders in a list and connect a produced pallet to a certain order. This is because the bakery should not have pallets in store that would have a possibility to go bad. This way pallets will only be produced if there is an order that requires the pallet to be produced.

2.6 Delivery

With only this part of the program all produced pallets are put in the freezer by default and no delivery is handled with any pallets. For implementing the delivery one could only move (change) the pallets location from freezer to delivered.

3 System outline

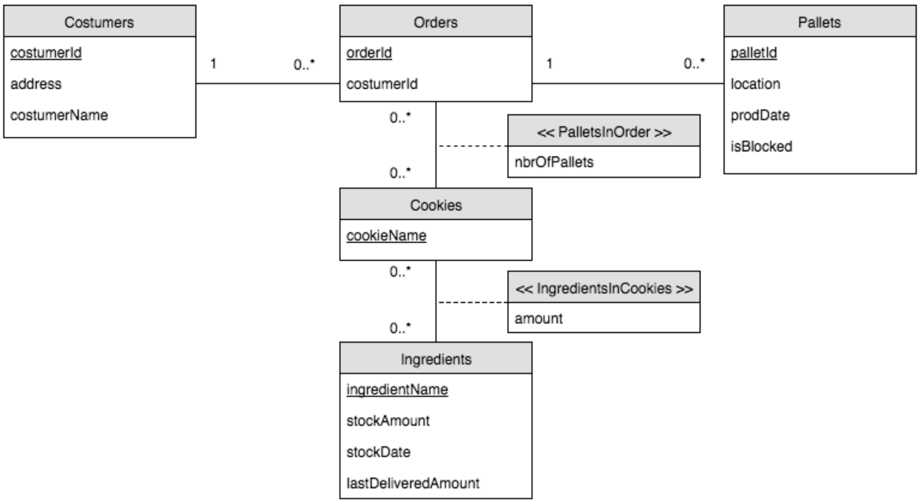
3.1 Database Manager

The application is using MYSQL as it communicates with the database through LTH's server.

3.2 Language

The program is implemented as an java application using Java Swing for implementing the interface where the client is interacting with the production.

4 ER - Diagram



5 Relations

Here is the relation diagram presented as implemented.

Foreign key is represented by *italics*
Primary key is represented by underline

Ingredients(ingredientName, stockAmount, stockDate, lastDeliverdAmount)
IngredientsInCookies(*cookieName*, *ingredientName*, amount)
Cookies(cookieName)
Costumers(costustumerId, address, costumerName)
Orders(orderId, oderDate, costumerId)
Pallets(palletNbr, prodDate, *cookieName*, location, isBlocked, *orderId*)
PalletsInOrders(*orderId*, *cookieName*, nrbOfPallets)

6 SQL Statements

```
create table Ingredients (
  ingredientName varchar(30) not null,
  stockAmount integer not null,
  stockDate date not null,
  lastDeliveredAmount integer not null,
  primary key(ingredientName)
);

create table IngredientsInCookies (
  cookieName varchar(30) not null,
  ingredientName varchar(30) not null,
  amount integer not null,
  primary key(cookieName,ingredientName),
  foreign key(cookieName) references Cookies(cookieName),
  foreign key(ingredientName) references Ingredients(ingredientName)
);

create table Cookies (
  cookieName varchar(30)  not null,
  primary key(cookieName)
);

create table Customers (
  customerId integer auto_increment,
  address varchar(30) not null,
  customerName varchar(30) not null,
  primary key(customerID)
);

create table Orders  (
  orderId integer auto_increment,
  orderDate date not null,
  customerId integer not null,
  primary key(orderId),
  foreign key(customerId) references Customers(customerId)
);

create table Pallets (
  palletNbr integer auto_increment,
  cookieName varchar(30) not null,
  prodDate date not null,
  location varchar (30) DEFAULT 'Freezer',
  isBlocked varchar(10) DEFAULT 'false',
  orderId integer,
  primary key(palletNbr),
  foreign key(cookieName) references Cookies(cookieName),
  foreign key(orderId) references Orders(orderId)
);

create table PalletsInOrders  (
  orderId integer not null,
  cookieName varchar(30) not null,
  nbrOfPallets integer not null,
  primary key(orderId, cookieName),
  foreign key(orderId) references Orders(orderId),
  foreign key(cookieName) references Cookies(cookieName)
);
```

7 User Manual

The GUI is built up by 4 different views, namely: "*produce pallets*", "*Search Cookie Pallets*", "*Search Pallet number*", "*Search by date Pallets*".

The produce pallets"view lets the user create a pallet of a chosen cookie type. By first choosing a cookie and then press Produce Pallet". Then the pallet's information will be displayed in the view.

The Search Cookie Pallets"view lets the user search for all the pallets of a specified cookie type. It is also possible to block all those pallets of that cookie type by clicking on Block the chosen cookieänd follow by clicking Search". The information of pallets containing the searched cookie will be displayed.

The Search by date Pallets"view lets the user find all pallets for a cookie in an interval of production. By first clicking on a cookie and then type in Start date following by End date respectively and then press Search". The information about the pallets within the interval for a given cookie will be displayed.

The program will quit after closing down the window.