Databas

Alexander Israelsson F12 israelsson.alexander@gmail.com
Emil Westenius F12 emil@westenius.se
Marcus Kindberg π12 tpi12mki@student.lu.se

March 2016

1 Introduction

In this project we have created a database and a client for a baking company called Krusty Kookies Sweden AB. The focus was to provide a sample solution which can easily be expanded to be used throughout the company. In this initial solution the entire database was created with only a client managing the production, blocking and searching of pallets of cookies.

2 System outline

The database management system used was MySql and the client was implemented with Java. The UML diagram for the system can be viewed in figure 1

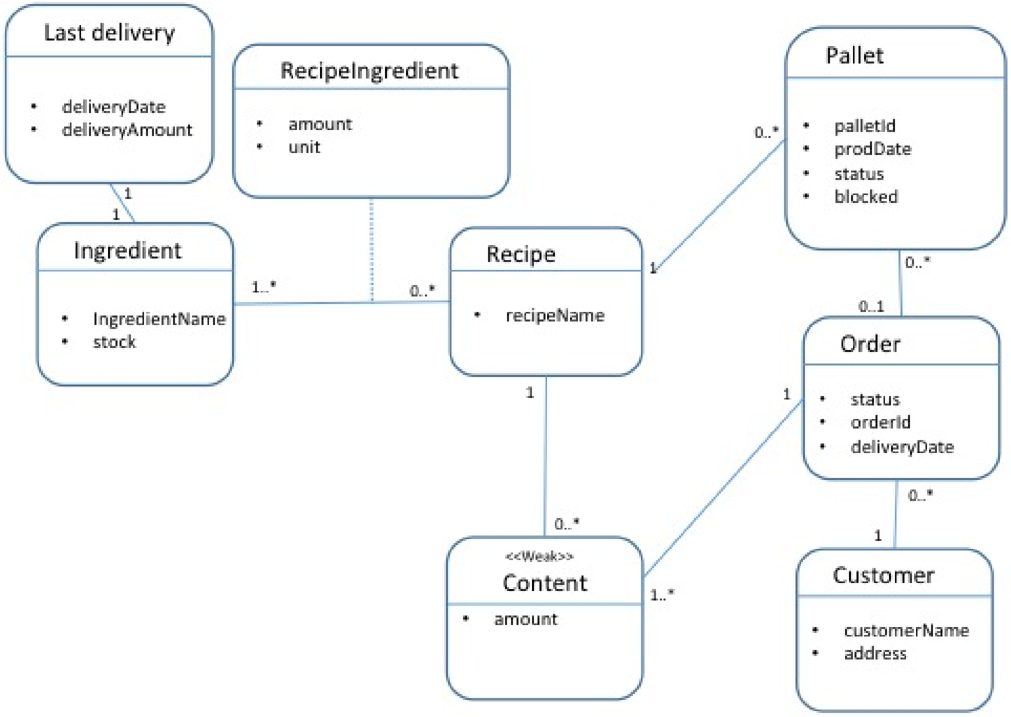

Figure 1: A UML diagram of the system.

The code for table generation can be seen in section 5. An example of the GUI can be observed in figure 2. In order to communicate with the database from the client the JDBC, Java DataBase Connectivity, API was used.

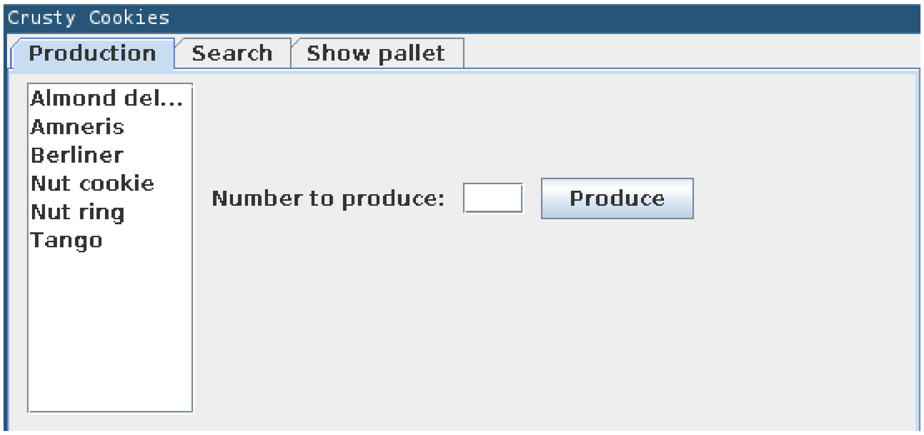

Figure 2: A view of the screen where cookies are produced.

In the client which was created one can produce pallets of cookies from recipes and ingredients in the database. The client allows for searching for pallets with date and cookie type as parameters, the found pallets can then be blocked. All information concerning a specific pallet can be viewed if the pallet number is given.

3 Requirements

As stated in the introduction we were to implement a database and a partially working client. This entailed that through the client being able handle everything concerning production, searching and blocking of pallets. The database itself needs to be fully implemented and having the tools to be able to for example trace pallets, track ingredient stock, customer orders and so on. All requirements above were met in this project, left to consider is to perhaps split the finished orders from the active ones. This should probably be done in order to keep performance good when searching for active orders since this is assumed to be more important than searching for finished orders.

4 Relations

Recipe(recipeName)

Ingredient(ingredientName,stock,deliveryDate,deliveryAmount)

RecipeName(*ingredientName*,*recipeName*,amount)

Pallet(palNbr,prodDate,*recipeName*,status,blocked)

DeliveredPallets(*orderId*,*palNbr*)

OrderContent(*orderId*,*recipeName*,amount)

Order(*customerName*,*address*, deliveryDate, status, orderId)

Customer(customerName,address)

All relations above are in BCNF SINCE REASONS

5 SQL code

The sql code for creating the tables and inserting the recipes, cookie types and ingredients can be viewed below.

```
set foreign_key_checks = 0;

drop table if exists Recipe;
drop table if exists Ingredient;
drop table if exists RecipeIngredient;
drop table if exists Pallet;
drop table if exists Orders;
drop table if exists DeliveredPallets;
drop table if exists OrderContent;
drop table if exists Customer;
set foreign_key_checks = 1;

create table Recipe(
  recipeName varchar(20),
  primary key(recipeName)
);

create table Ingredient(
  IngredientName varchar(25),
  stock int,
  deliveryDate date,
  deliveryAmount int,
  primary key(IngredientName)
);

create table RecipeIngredient(
  recipeName varchar(20),
  IngredientName varchar(25),
  amount decimal(5, 2),
  unit varchar(20),
  primary key(IngredientName, recipeName),
  foreign key(IngredientName) references Ingredient(IngredientName),
  foreign key(recipeName) references Recipe(recipeName)
);
```

```
create table Pallet(
  palletId int AUTO_INCREMENT,
  prodDate datetime,
  recipeName varchar(20),
  status varchar(10) DEFAULT ”DEEPFREEZE”,
  blocked boolean DEFAULT False,
  primary key(palletId),
  foreign key(recipeName) references Recipe(recipeName)
);

create table Orders(
  orderId int AUTO_INCREMENT,
  status varchar(10),
  deliveryDate date,
  primary key(orderId)
);

create table OrderContent(
  orderId int ,
  type varchar(20),
  amount int,
  primary key(orderId, type),
  foreign key(orderId) references Orders(orderId)
);
create table DeliveredPallets(
  orderId int,
  palletId int,
  primary key(orderId, palletId),
  foreign key(orderId) references Orders(orderId),
  foreign key(palletId) references Pallet(palletId)
);

create table Customer(
  customerName varchar(20),
  adress varchar(20),
  primary key(customerName, adress)
);

insert into Recipe values (”Nut ring”);
insert into Recipe values (”Nut cookie”);
insert into Recipe values (”Amneris”);
insert into Recipe values (”Almond delight”);
insert into Recipe values (”Berliner”);
insert into Recipe values (”Tango”);
insert into Ingredient values (”Flour”,100000,CURDATE(),100000);
insert into Ingredient values (”Butter”,100000,CURDATE(),100000);
insert into Ingredient values (”Eggs”,10000,CURDATE(),10000);
insert into Ingredient values (”Icing sugar”,10000,CURDATE(),10000);
insert into Ingredient values (”Roasted, chopped nuts”,10000,CURDATE(),10000);
insert into Ingredient values (”Fine−ground nuts”,10000,CURDATE(),10000);
insert into Ingredient values (”Ground, roasted nuts”,10000,CURDATE(),10000);
insert into Ingredient values (”Bread crumbs”,10000,CURDATE(),10000);
insert into Ingredient values (”Sugar”,100000,CURDATE(),100000);
insert into Ingredient values (”Egg whites”,1000,CURDATE(),1000);
insert into Ingredient values (”Chocolate”,100000,CURDATE(),100000);
insert into Ingredient values (”Marzipan”,100000,CURDATE(),100000);
insert into Ingredient values (”Potato starch”,10000,CURDATE(),10000);
insert into Ingredient values (”Wheat flour”,10000,CURDATE(),10000);
insert into Ingredient values (”Sodium bicarbonate”,10000,CURDATE(),10000);
insert into Ingredient values (”Vanilla”,10000,CURDATE(),10000);
insert into Ingredient values (”Chopped almonds”,10000,CURDATE(),10000);
insert into Ingredient values (”Cinnamon”,10000,CURDATE(),10000);
insert into Ingredient values (”Vanilla sugar”,10000,CURDATE(),10000);


insert into RecipeIngredient values(”Nut ring”,”Flour”,450,”g”);
insert into RecipeIngredient values(”Nut ring”,”Butter”,450,”g”);
insert into RecipeIngredient values(”Nut ring”,”Icing sugar”,190,”g”);
insert into RecipeIngredient values(”Nut ring”,”Roasted, chopped nuts”,225,”g”);

insert into RecipeIngredient values(”Nut cookie”,”Fine−ground nuts”,750,”g”);
insert into RecipeIngredient values(”Nut cookie”,”Ground, roasted nuts”,625,”g”)
insert into RecipeIngredient values(”Nut cookie”,”Bread crumbs”,125,”g”);
insert into RecipeIngredient values(”Nut cookie”,”Sugar”,375,”g”);
insert into RecipeIngredient values(”Nut cookie”,”Egg whites”,3.5,”dl”);
insert into RecipeIngredient values(”Nut cookie”,”Chocolate”,50,”g”);

insert into RecipeIngredient values(”Amneris”,”Marzipan”,750,”g”);
insert into RecipeIngredient values(”Amneris”,”Butter”,250,”g”);
insert into RecipeIngredient values(”Amneris”,”Eggs”,250,”g”);
insert into RecipeIngredient values(”Amneris”,”Potato starch”,25,”g”);
insert into RecipeIngredient values(”Amneris”,”Wheat flour”,25,”g”);

insert into RecipeIngredient values(”Tango”,”Butter”,200,”g”);
insert into RecipeIngredient values(”Tango”,”Sugar”,250,”g”);
insert into RecipeIngredient values(”Tango”,”Flour”,300,”g”);
insert into RecipeIngredient values(”Tango”,”Sodium bicarbonate”,4,”g”);
insert into RecipeIngredient values(”Tango”,”Vanilla”,2,”g”);

insert into RecipeIngredient values(”Almond delight”,”Butter”,400,”g”);
insert into RecipeIngredient values(”Almond delight”,”Sugar”,270,”g”);
insert into RecipeIngredient values(”Almond delight”,”Chopped almonds”,279,”g”);
insert into RecipeIngredient values(”Almond delight”,”Flour”,400,”g”);
insert into RecipeIngredient values(”Almond delight”,”Cinnamon”,10,”g”);

insert into RecipeIngredient values(”Berliner”,”Flour”,350,”g”);
insert into RecipeIngredient values(”Berliner”,”Butter”,250,”g”);
insert into RecipeIngredient values(”Berliner”,”Icing sugar”,100,”g”);
insert into RecipeIngredient values(”Berliner”,”Eggs”,50,”g”);
insert into RecipeIngredient values(”Berliner”,”Vanilla sugar”,5,”g”);
insert into RecipeIngredient values(”Berliner”,”Chocolate”,50,”g”);
```