Krusty Kookies AB Project Report

2015-03-30

Kit Gustavsson, Data, kitgustavsson@gmail.com
Fredrik Hagfjäll, InfoCom, dic12fha@student.lu.se
Erik Stenlund, Data, zba10est@student.lu.se
Olof Wahlgren, Data, ama09owa@student.lu.se

Introduction

The project aim was to develop and deliver a working database with GUI to Krusty Kookies AB. The entire database was implemented, but the program only supports managing of production, search and blocking of pallets. Our project does not handle anything concerning raw materials, recipes, orders or delivery, except for updating the raw material storage when new pallets are produced.

Requirements

The implementation fulfills all the requirements given. These do not specify explicitly what information to show and what not to, so a universal view of pallets was implemented. It shows information which differs from pallets of different states, and allows the user to get all the information explicitly mentioned in the requirements.

System

Our system relies on a PHP-server which communicates with a DBMS that is of MySQL type. We are using a MVC layout and the file database.inc.php handles all the communication with the database. We use `puccini.cs.lth.se` as our database-server. As the web pages need date and time input, we are using HTML5. This requires the use of a fairly new web browser which supports HTML5.

E/R & SQL

An E/R-model of the database can be found under Appendix A, subsection E/R. The SQL statements that creates all the tables can be found under Appendix A, subsection SQL.

Relations

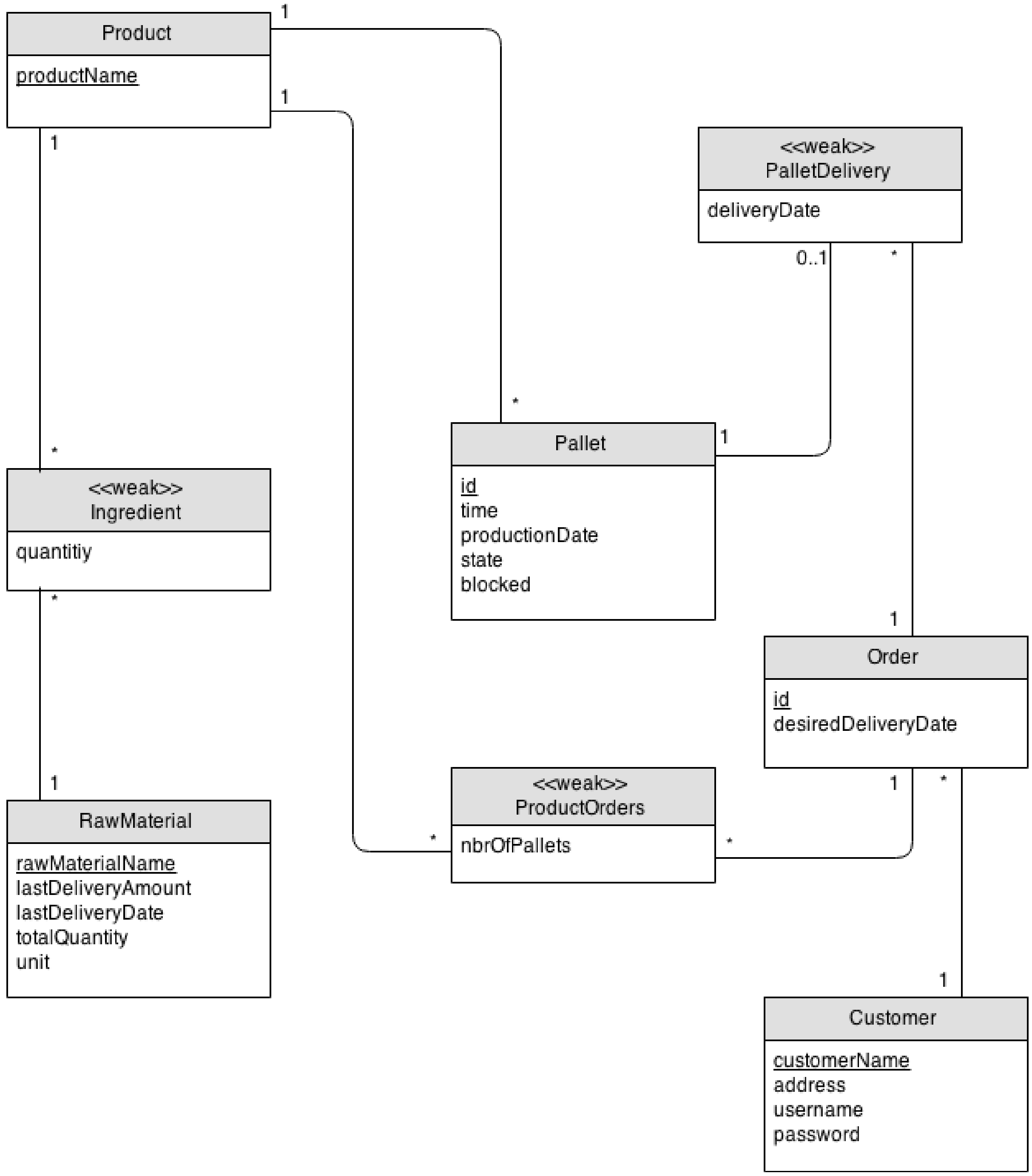
Primary keys are underlined and foreign keys in *italic*.

RawMaterials (rawMaterialName, lastDeliveryDate, lastDeliveryAmount, totalQuantity, unit)
Products(productName)
Ingredients(*ingredientName, productName*, quantity);
Customers(customerName, address, username, password)
Orders(orderId, desiredDeliveryDate, *customerName*)
Pallets(palletId, productionDateTime, state, blocked, *productName*)
PalletDeliveries(*palletId, orderId*, deliveryDateTime)
ProductOrders(*orderId, productName*, nbrOfPallets)

All relations are in BCNF except for `Customers` since the only functional dependencies is key-dependencies. `Customers` is not in BCNF since `username` is not a key, but the functional dependency that `username` gives all other attributes in the relation is not used. The `Customer` relation is in 3NF though, since all the attributes on the right hand side of all the functional dependencies is a key or part of one.

Appendix A

E/R



SQL

```
create table RawMaterials (
rawMaterialName varchar(30),
lastDeliveryDate date NOT null,
lastDeliveryAmount int NOT null,
totalQuantity double NOT null,
unit varchar(30) NOT null,
primary key(rawMaterialName)
);

create table Products (
productName varchar(30),
primary key(productName)
);

create table Ingredients (
ingredientName varchar(30),
productName varchar(30),
quantity double NOT null,
primary key (ingredientName, productName),
foreign key (ingredientName) references RawMaterials(rawMaterialName),
foreign key (productName) references Products(productName)
);

create table Customers (
customerName varchar(30),
address varchar(30),
username varchar(30),
password varchar(30),
primary key (customerName)
);

create table Orders (
orderId int auto_increment,
desiredDeliveryDate date NOT null,
customerName varchar(30) NOT null,
primary key(orderId),
foreign key (customerName) references Customers(customerName)
);

create table Pallets (
palletId int auto_increment,
productionDateTime datetime NOT null DEFAULT NOW(),
state varchar(30) NOT null DEFAULT 'freezer',
blocked bool NOT null DEFAULT false,
productName varchar(30) NOT null,
primary key (palletId),
foreign key (productName) references Products(productName)
);

create table PalletDeliveries (
palletId int,
orderId int,
deliveryDateTime datetime NOT null,
primary key (palletId, orderId),
foreign key (palletId) references Pallets(palletId),
foreign key (orderId) references Orders(orderId)
);

create table ProductOrders (
orderId int,
productName varchar(30),
nbrOfPallets int NOT null,
primary key (orderId, productName),
foreign key (orderId) references Orders(orderId),
foreign key (productName) references Products(productName)
);
```