

# EDA216 Database Technology Project

## Cookie production and delivery management system

Johan Gran Nilsson  
Dept. of Comp. Science, Lund University  
Email: [ada10jgr@student.lu.se](mailto:ada10jgr@student.lu.se)

Huang Li  
Dept. of Comp. Science, Lund University  
Email: [int15lh1@student.lu.se](mailto:int15lh1@student.lu.se)

Magnus Törnquist  
Dept. of Comp. Science, Lund University  
Email: [magnusmvt@gmail.com](mailto:magnusmvt@gmail.com)

Submission date: 2015-3-30

## Introduction

This project is about modeling and implementing a database for cookie production management for Krusty Kookies Sweden AB (KKS AB). The system is divided into three parts:

1) Raw material management;

The system should be able to auto-update and record the remaining amount of ingredients after production.

2) Producing, blocking and searching functions;

The production quantity is based on following week's orders, the unit of production is pallet, each pallet contains identifiable information such as pallet number, cookie name, producing date, etc.

Pallets can be blocked with product name and producing dates.

Also, there are several search functions for the convenience of database checking.

3) Order and delivery management.

Customers can place order and delivery date can be checked in the system.

In the above programs, section 2 is implemented and also place order in section 3.

## Project requirements

### Production

- Be able to check how many pallets of a product that have been produced during a specific time.

### Raw Materials

- Be able to update the raw materials automatically after production
- Be able to check the amount in store of each ingredient
- Be able to check when, and how much of, an ingredient was last delivered into storage.

### Recipes

- Have an interface to show the collection of recipes
- Have a facility for entering new recipes.

### Produced Pallets

- Be able to see all information about a pallet with a given number (the contents of the pallet, the location of the pallet, if the pallet is delivered and in that case to whom, etc.).
- Be able to see which pallets that contain a certain product and which pallets that have been produced during a certain time interval.
- Be able to find out which products that are blocked, and also which pallets that contain a certain blocked product.

- Be able to check which pallets that have been delivered to a given customer, and the date and time of delivery.

### Orders and Production Planning

- Orders must be registered in the database.
- Have a facility to see all orders that are to be delivered during a specific time period.

### Delivery

- Be able to update delivered pallets' data with customer information and date of delivery.

## Project outline

In this project we use MySQL as the database manager and PHP as the interface. The database in MySQL contains necessary tables and initial data for modeling. The functions which communicate with database is in *database.inc.php* file, while other php files constructs the interfaces for loading or showing information.

## E/R diagram

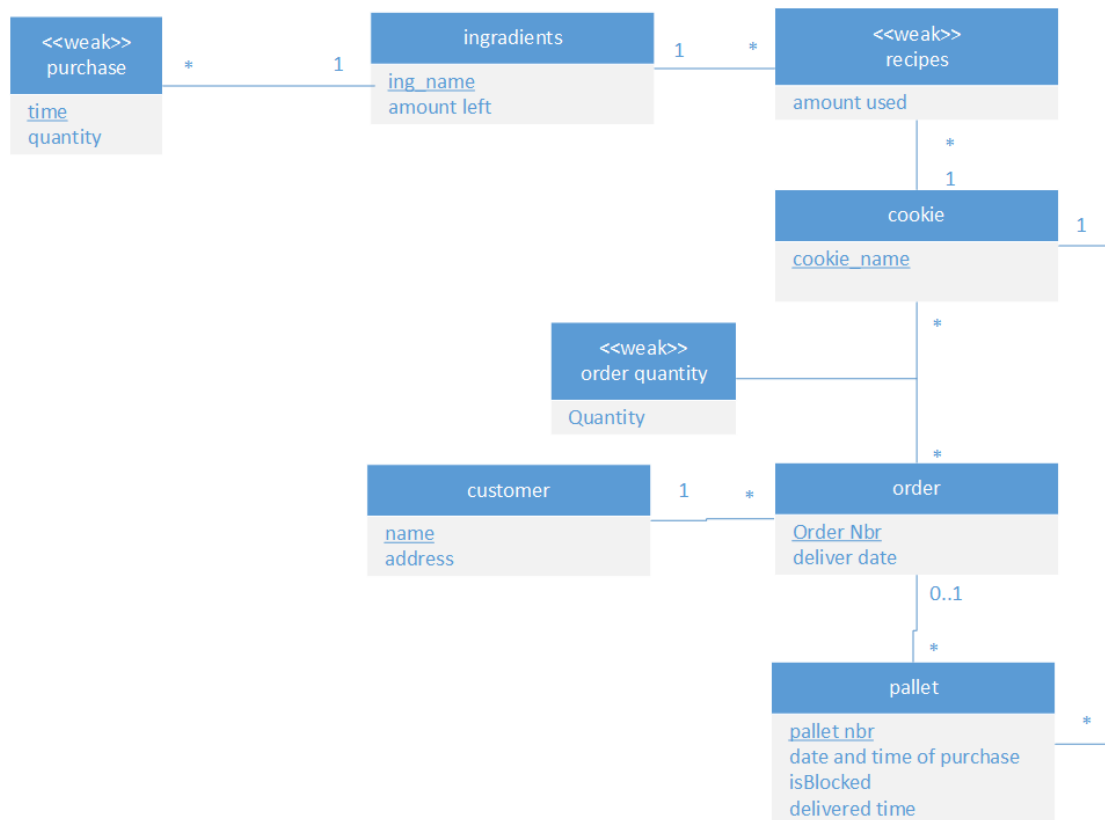


Figure 1: UML diagram for production management system

## Relations

ingredient(ingredient\_name, amount\_left) :  
BCNF: ingredient\_name → amount\_left

cookie(cookie\_name) : BCNF

order(order\_nbr, *customer\_name*, deliver\_date):  
BCNF: order\_nbr → *customer\_name*, deliver\_date

customer(customer\_name, address):  
BCNF: customer\_name → address

pallets (pallet\_nbr, *cookie\_name*, purchase\_datetime, isBlocked, delivered\_time):  
BCNF: pallet\_nbr → *cookie\_name*, purchase\_datetime, isBlocked, delivered\_time

ingre\_purchase(ingredient\_name, time, quantity):  
BCNF: ingredient\_name, time → quantity

recipe (cookie\_name, ingredient\_name, amount\_used):  
BCNF: cookie\_name, ingredient\_name → amount\_used

order\_quantity(cookie\_name, order\_nbr, quantity):  
BCNF: cookie\_name, order\_nbr → quantity

## SQL statements

```
CREATE TABLE Customer (  
    name VARCHAR(50) NOT NULL,  
    location VARCHAR(50) NOT NULL,  
    PRIMARY KEY (name)  
);
```

```
CREATE TABLE Cookie (  
    name VARCHAR(50) NOT NULL,  
    PRIMARY KEY (name)  
);
```

```
CREATE TABLE Ingredient (  
    name VARCHAR(50) NOT NULL,  
    amount INT,  
    PRIMARY KEY (name)  
);
```

```
CREATE TABLE Orders (  
    name VARCHAR(50) NOT NULL,  
    amount INT,  
    PRIMARY KEY (name)  
);
```

```

    orderNbr INT auto_increment,
    customerName VARCHAR(50) NOT NULL,
    deliveryDate DATE NOT NULL,
    PRIMARY KEY (orderNbr),
    FOREIGN KEY (customerName) REFERENCES Customer(name)
);

CREATE TABLE Recipe (
    cookieName VARCHAR(50) NOT NULL,
    ingredientName VARCHAR(50) NOT NULL,
    amountUsed INT,
    FOREIGN KEY (cookieName) REFERENCES Cookie(name),
    FOREIGN KEY (ingredientName) REFERENCES Ingredient(name)
);

CREATE TABLE IngredientPurchase (
    ingredientName VARCHAR(50) NOT NULL,
    time DATE NOT NULL,
    quantity INT,
    FOREIGN KEY (ingredientName) REFERENCES Ingredient(name)
);

CREATE TABLE OrderQuantity (
    cookieName VARCHAR(50) NOT NULL,
    orderNbr INT NOT NULL,
    quantity INT,
    FOREIGN KEY (cookieName) REFERENCES Cookie(name),
    FOREIGN KEY (orderNbr) REFERENCES Orders(orderNbr)
);

CREATE TABLE Pallet (
    palletID INT auto_increment,
    cookieName VARCHAR(50) NOT NULL,
    productionDate DATE,
    deliveredDate DATE,
    isBlocked INT NOT NULL,
    orderNbr INT,
    PRIMARY KEY (palletID),
    FOREIGN KEY (cookieName) REFERENCES Cookie(name),
    FOREIGN KEY (orderNbr) REFERENCES Orders(Ordernbr)
);

```