

**Bavarder’s Pizza Datenbank Projekt**

Teammitglieder: S.Hajnik, O.Achermann, R.Blaauw

Lehrperson: G.Gesell

Modul 153

Inhalt

[1 Einführung 2](#_Toc61356421)

[2 Informieren 2](#_Toc61356422)

[2.1 Inspiration 2](#_Toc61356423)

[2.2 Tutorials 2](#_Toc61356424)

[3 Planen 2](#_Toc61356425)

[3.1 Kanban 2](#_Toc61356426)

[4 Entscheiden 3](#_Toc61356427)

[4.1 Entscheidungsmatrix 3](#_Toc61356428)

[5 Realisieren 4](#_Toc61356429)

[5.1 Key und Tempo definieren 4](#_Toc61356430)

[5.2 Chords schreiben 4](#_Toc61356431)

[5.3 Melodie schreiben 4](#_Toc61356432)

[5.4 Drums programmieren 5](#_Toc61356433)

[5.5 Samples suchen und einbauen 6](#_Toc61356434)

[5.6 Bassline schreiben 6](#_Toc61356435)

[5.7 Arrangement erstellen 6](#_Toc61356436)

[5.8 Mix und Mastering erstellen 6](#_Toc61356437)

[6 Kontrollieren 8](#_Toc61356438)

[6.1 Kontrollmatrix 8](#_Toc61356439)

[6.2 Resultat 8](#_Toc61356440)

[7 Auswerten 8](#_Toc61356441)

[7.1 Auswertungsmatrix 8](#_Toc61356442)

# Problembeschreibung

Für eine fiktive Pizzeria wird eine Datenbank erstellt. Die Pizzeria hat einen hauseigenen Lieferdienst und diverse Zutaten für ihre Pizzen. Stammkunden möchte man persistieren und nicht immer wieder neu anlegen müssen. Es gibt Menupizzen, aber auch die Möglichkeit eine eigene Pizza zusammen zu stellen.

# Anforderungsanalyse

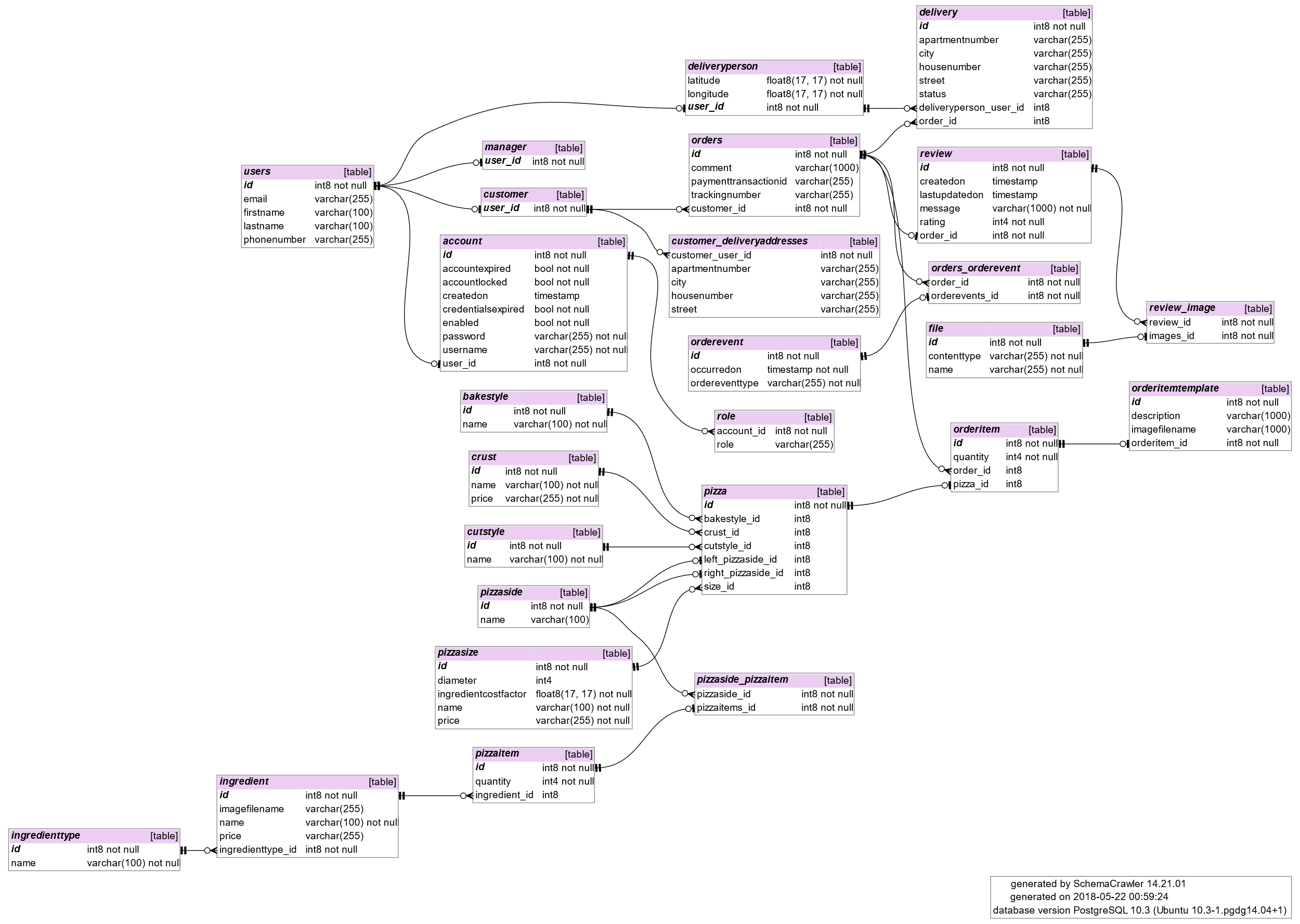
Folgende MUSS-Ziele sind gewünscht:

* Kunden anlegen
* Menupizzen abbilden
* Alles Pizzazutaten speichern
* Lieferaufträge annehmen
* Adressen anlegen
* Zutatenlieferanten einsehen

# Informieren

## Inspiration

Für Inspiration für unser Pizzeria, suchten wir die gängigen Lieferdienste und Pizza Take-Aways auf (Eat.ch, Dominos oder Dieci’s). Wir notierten uns mögliche Entitäten wie Zutaten, Pizzagrösse, Bestellung oder Kunde bzw. Person und diskutierten bereits Normalisierung der einzelnen Elemente unserer Datenbank. Via Github fanden wir auch eine komplexe Pizzeria Datenbank mit sehr vielen Entitäten und Material für Ideen für unser Projekt. Unten eine Abbildung des ERM von diesem Projekt.



# Planen

## Kanban

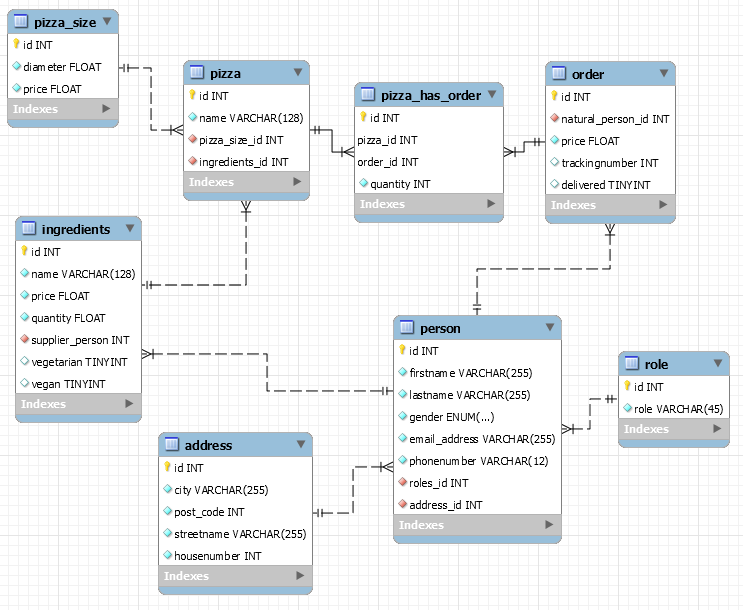
Für die Planung von unserem Projekt verwendeten wir Kanban. Auf einem Blick sehen wir wen an was Arbeitete und was zu erledigen ist bzw. was bereits erledigt wurde. Unten ein Screenshot unseres Kanbans.

Ein Bild, das Text, Screenshot, Monitor, schwarz enthält.

Automatisch generierte BeschreibungIn diesem System mussten wir nichts anderes machen als am Dienstagnachmittag das Kanban zu öffnen und unsere Aufgaben für das entsprechende Datum entgegenzunehmen bzw. am Schluss abzuschliessen.

# Realisieren

## ERM



Bei unserem ERM ist natürlich die Tabelle «Pizza» im Zentrum. Wobei die Grösse und die Zutaten ausgelagert sind. Die Namen der Menüpizzen kann man jedoch in der Tabelle Pizza speichern. In der Tabelle «Order» werden die Bestellungen gespeichert. Mit einer Transformationstabelle wird die «mc zu mc»-Beziehung zwischen «Pizza» und «Order» aufgelöst. In der Tabelle «Order» werden auch Status wie «Trackingnumber» oder «Delivered» festgehalten. Die Tabelle «Person» haben wir so weit wie gewünscht Normalisiert. Ihr Adressen sind ausgelagert und auch Rollen wird als Fremdschlüssel gesetzt. In der Tabelle Zutaten wird kommt eine Person ebenfalls als Fremdschlüssel vor, nämlich der Lieferant.

Mit diesem ERM sind die Wichtigsten Daten von den verschiedenen Pizzen bis hin zur Bestellungsabwicklung möglich. Auch Hierarchien wie «Manager-Employee» Beziehung lassen sich dank Rollenzuteilung abbilden.

## Testdaten generieren

# Kontrollieren

## Testprotokoll

Das Testing wurde an unsere Anforderungsanalyze angelehnt und die MUSS-Ziele werden kontrolliert.

**Funktionstests**

* **Mit Funktionellem Insert Statement**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case | Testing | Erwartet | Tatsächlich | OK/NOK |
| Kunden anlegen | Ich lege einen Benutzer an mit Insert Statement und wähle die Rolle „Client“ | Wird erstellt | Wird erstellt | OK |
| Menupizzen abbilden | Ich erstelle einen Benutzer an, die Pizzen werden angezeigt | Wird erstellt | Wird erstellt | OK |
| Pizzazutaten speichern | Ich erstelle meine eigene Pizza mit Insert Statement | Wird erstellt | Wird erstellt | OK |
| Lieferaufträge annehmen | Ich lege einen Benutzer an mit der Rolle „Employee“ mit Insert Statement | Wird erstellt | Wird erstellt | OK |
| Adressen anlegen | Ich lege einen Benutzer an mit Insert Statement | Wird erstellt | Wird erstellt | OK |
| Zutatenlieferanten einsehen | Ich lege einen Benutzer an mit Insert Statement und wähle die Rolle „Manager“ | Wird erstellt | Wird erstellt | OK |

* **Ohne Funktionellem Insert Statement**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Case | Testing | Erwartet | Tatsächlich | OK/NOK |
| Kunden anlegen | Ich lege einen Benutzer an mit falschen Daten an | Wird nicht erstellt | Wird nicht erstellt | OK |
| Menupizzen abbilden | Ich erstelle eine Benutzer an mit fehlerhaften Daten, und dadurch werden keine Pizzen angezeigt | Wird nicht erstellt | Wird nicht erstellt | OK |
| Pizzazutaten speichern | Ich erstelle meine eigene Pizza mit fehlerhaften Daten | Wird nicht erstellt | Wird nicht erstellt | OK |
| Lieferaufträge annehmen | Ich lege einen Benutzer an mit falschen Daten | Wird nicht erstellt | Wird nicht erstellt | OK |
| Adressen anlegen | Ich lege einen Benutzer an mit fehlerhaften Daten | Wird nicht erstellt | Wird nicht erstellt | OK |
| Zutatenlieferanten einsehen | Ich lege einen Benutzer an mit fehlerhaften Daten | Wird nicht erstellt | Wird nicht erstellt | OK |

# Auswerten

## Auswertungsmatrix

|  |  |
| --- | --- |
| ☺ (Was lief gut?)  Professioneller Sound  Gute Soundqualität  Gute Meilensteine(Melodie, Schlagzeug, Basslinie…)  Klare Aufteilung | 👎(Was lief weniger gut?)  Lange Ideenfindung  Verwendung von 2 verschiedenen Programmen |
| 🏱 (Was machen wir nächstes Mal besser?)  Schnellere Entscheidungen treffen  Nicht zu hohe Erwartungen | |