

Case Study

“Pizza bestellen”

Web Site:

www.soebes.com

Email:

training@soebes.com

Dipl.Ing.(FH) Karl Heinz Marbaise

Agenda

- 1.Datenbank Design
- 2.Klassen Diagramm
- 3.User Interface
- 4.SQL Abfrage

1. Datenbank Design

Übersicht

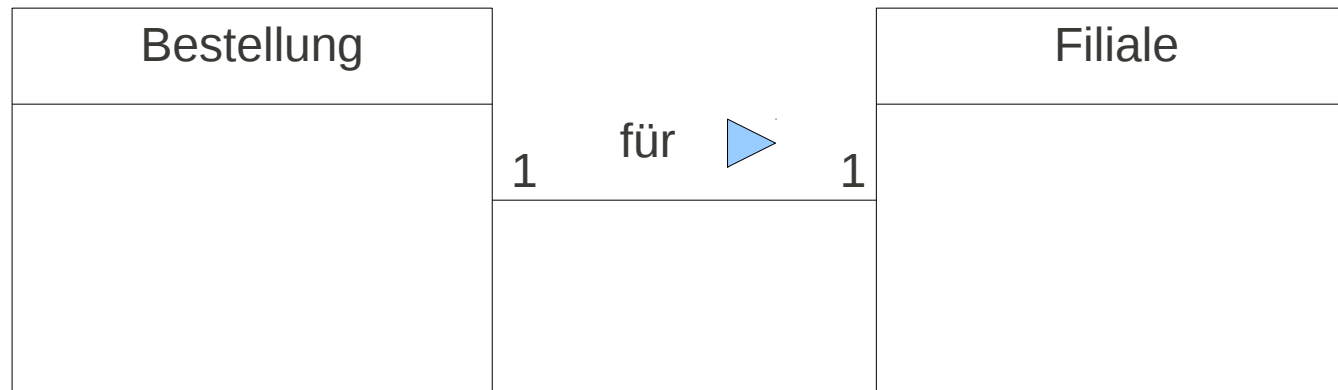
- Filiale
 - Die Filiale besteht aus Angaben zur Anschrift etc.
 - Aus den Anforderungen ergibt sich, dass mehrere Filialen existieren.
- Kunde
 - Name, Anschrift, Telefonnummer etc.
- Pizza
 - Eine Pizza besteht aus der gewählten Größe und einer Menge von Zutaten

1. Datenbank Design Übersicht

- Bestellung
 - Kunde, Filiale, Pizza, Zeitpunkt der Bestellung

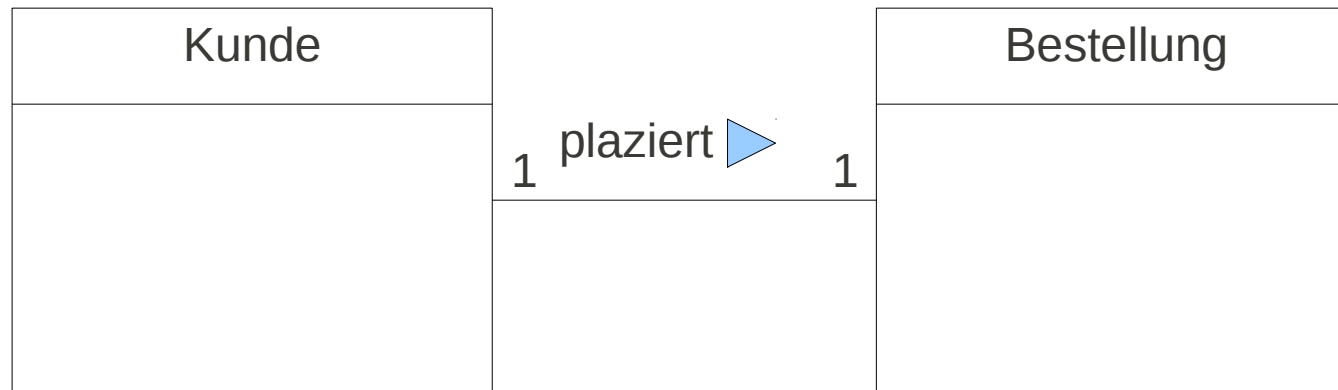
1. Datenbank Design Beziehungen

- Filiale zur Bestellung



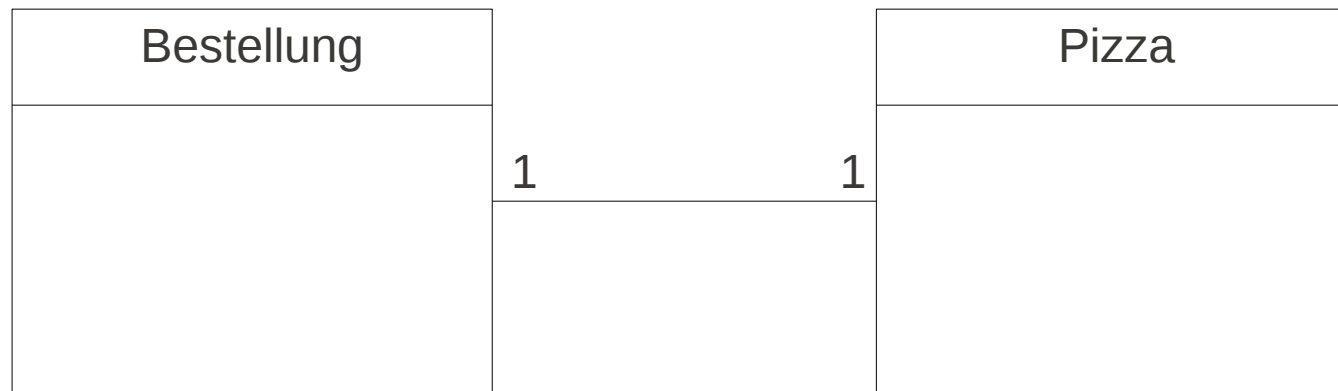
1. Datenbank Design Beziehungen

- Kunde zur Bestellung



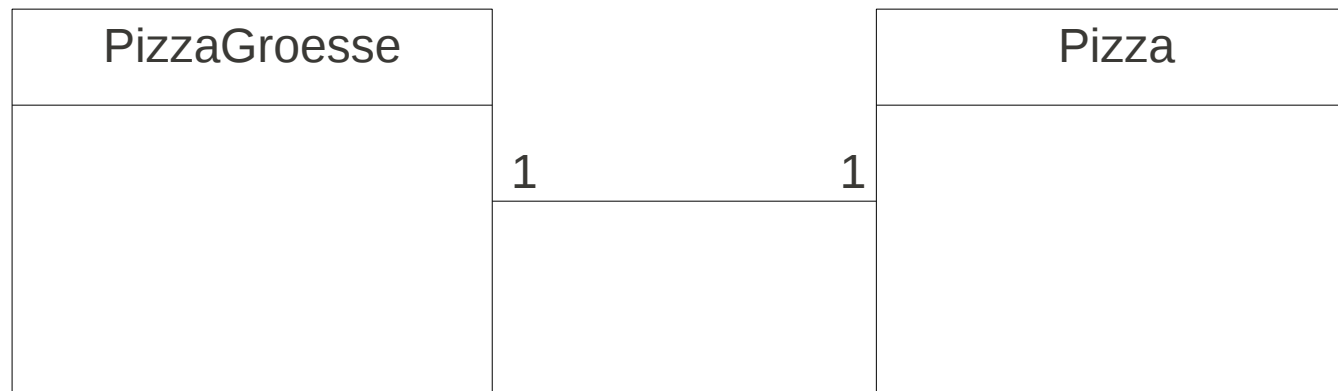
1. Datenbank Design Beziehungen

- Bestellung zur Pizza



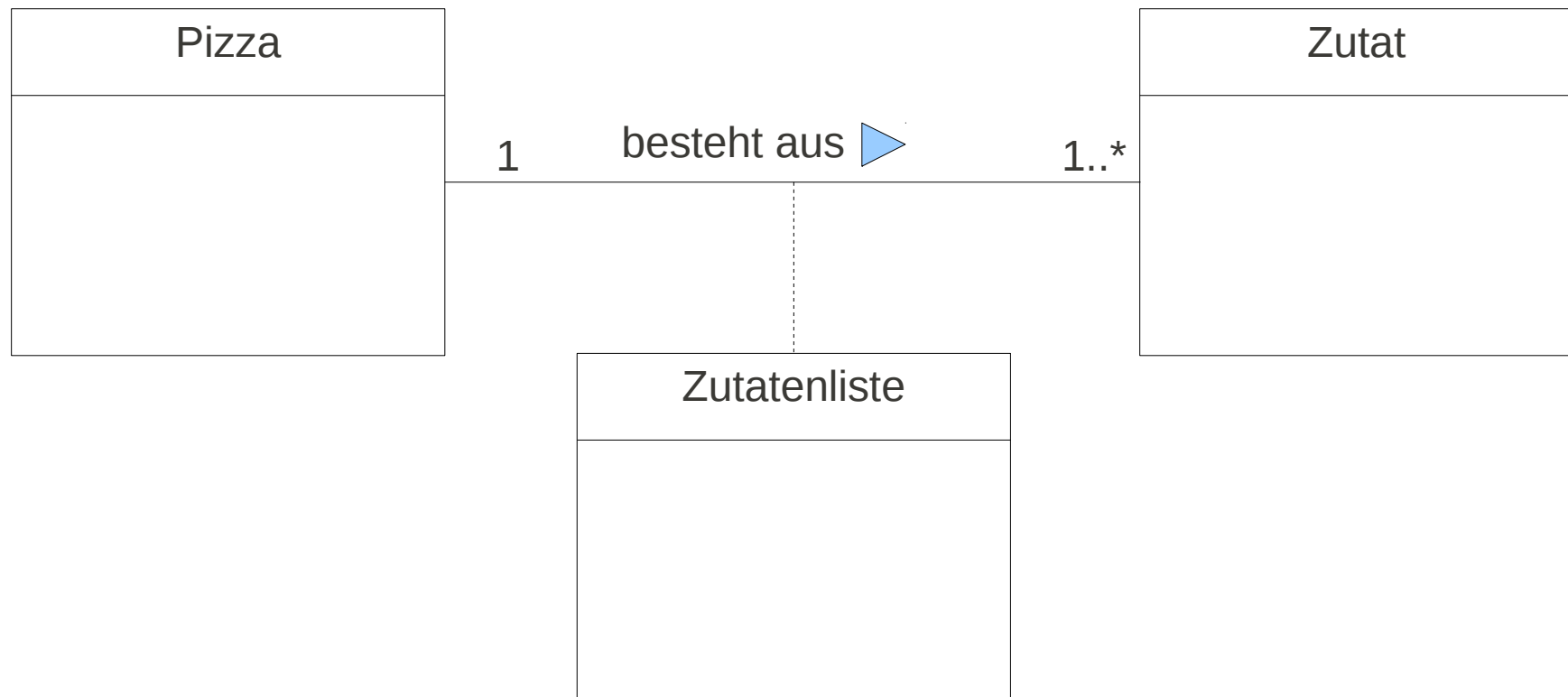
1. Datenbank Design Beziehungen

- Pizza zur Pizza Größe

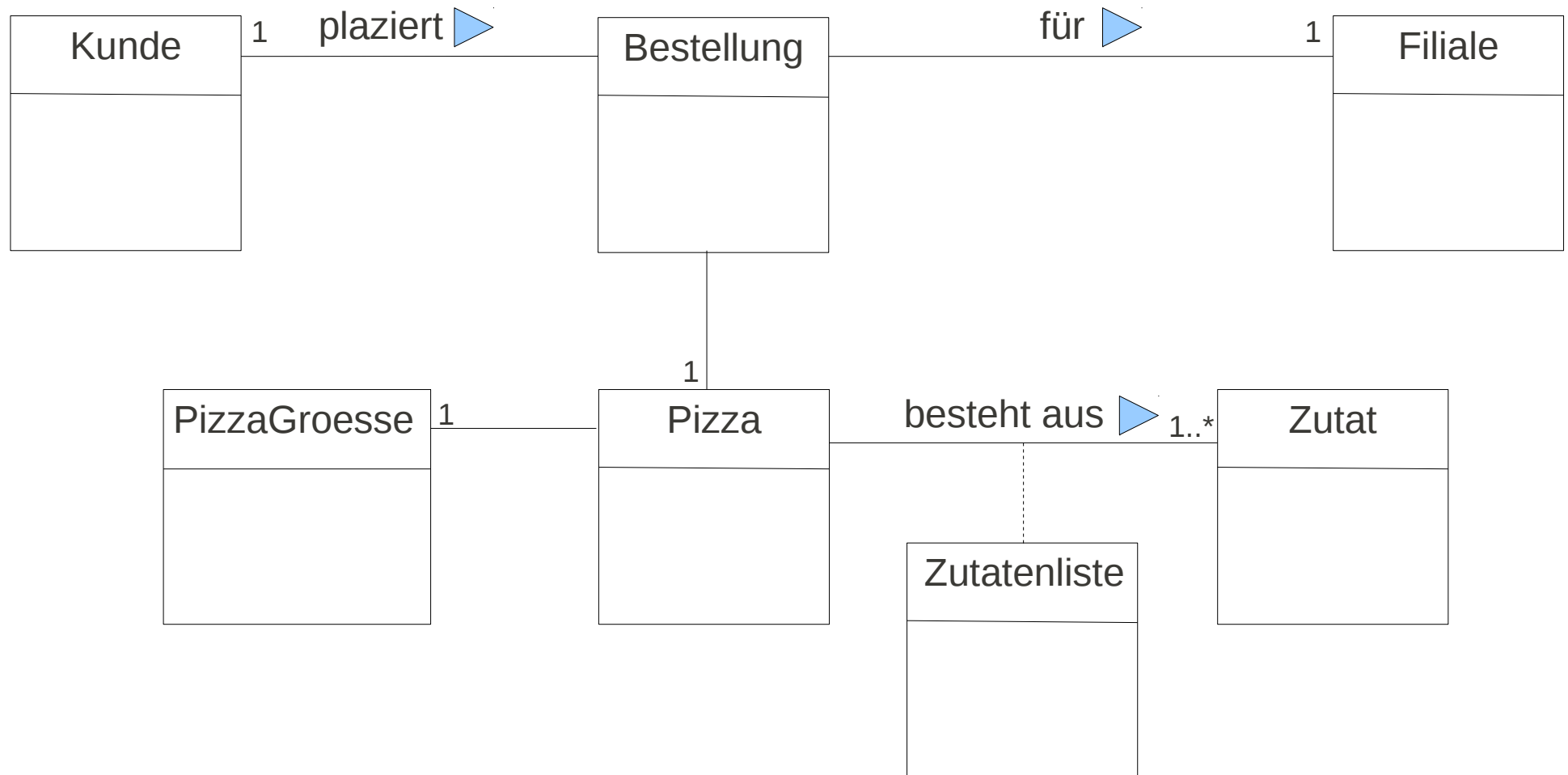


1. Datenbank Design Beziehungen

- Pizza zu Zutaten



1. Datenbank Design Übersicht



1. Datenbank Design

Attribute der Tabellen

- Tabelle: Filiale
 - Id
 - Technischer Schlüssel (auto_increment bzw. Sequence)
 - Name
 - varchar(255)

1. Datenbank Design

Attribute der Tabellen

- Tabelle: Zutat
 - Id
 - Technischer Schlüssel (auto_increment bzw. Sequence)
 - Name
 - varchar(255)

1. Datenbank Design

Attribute der Tabellen

- Tabelle: PizzaGroesse
 - Id
 - Technischer Schlüssel (auto_increment bzw. Sequence)
 - Name
 - varchar(255)

1. Datenbank Design

Attribute der Tabellen

- Tabelle: Kunde
 - Id
 - Technischer Schlüssel (auto_increment bzw. Sequence)
 - Name
 - varchar(255)
 - Adresse
 - varchar(255)
 - Telefonnummer
 - varchar(255)

1. Datenbank Design

Attribute der Tabellen

- Tabelle: Pizza
 - Id
 - Technischer Schlüssel (auto_increment bzw. Sequence)
 - fk_pizza_groesse
 - Fremd Schlüssel auf Tabelle PizzaGroesse

1. Datenbank Design

Attribute der Tabellen

- Tabelle: ZutatenListe
 - pizza_id
 - Fremd Schlüssel
 - zutaten_list_id
 - Fremd Schlüssel auf Tabelle Zutat

1. Datenbank Design

Attribute der Tabellen

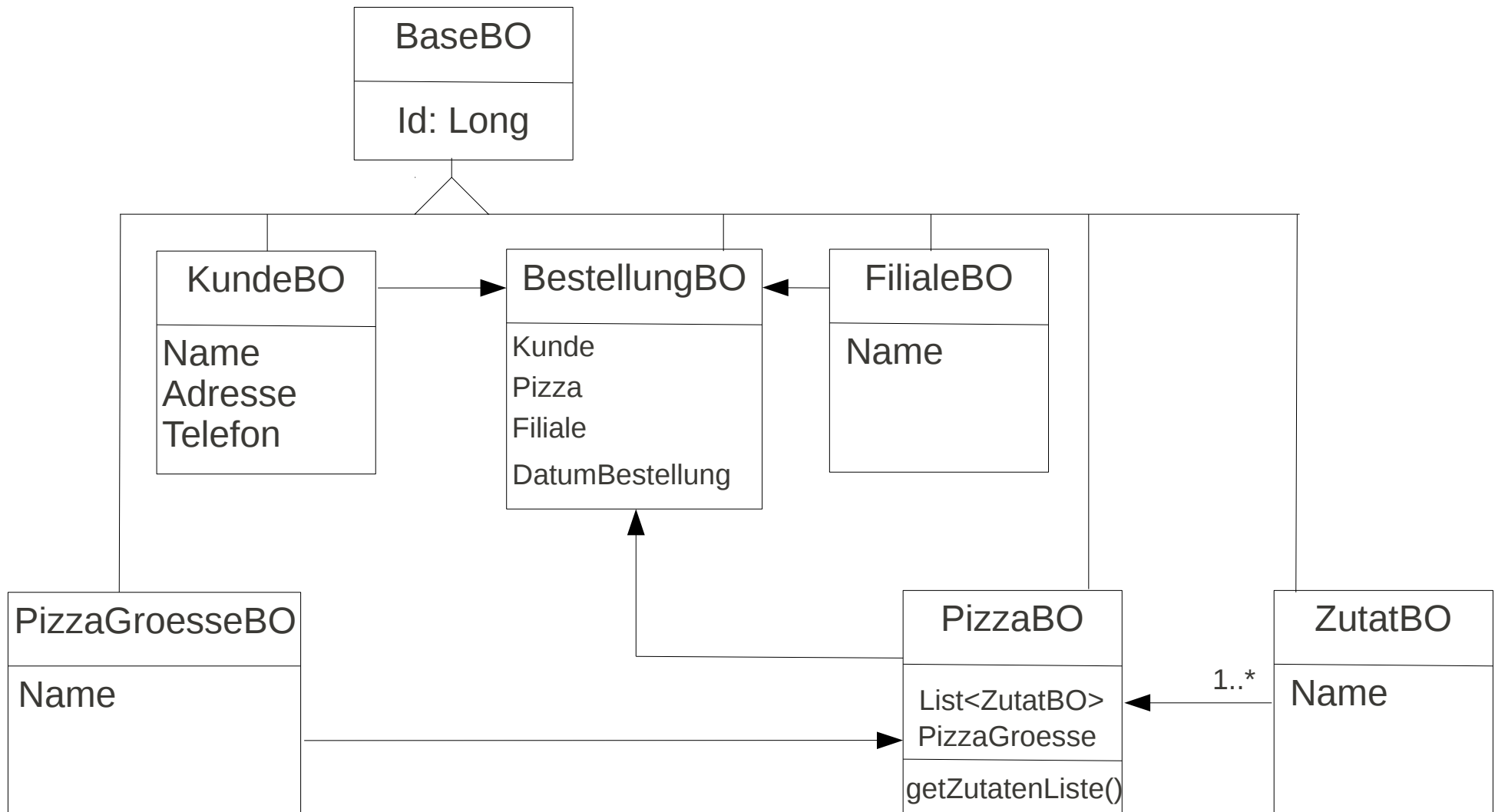
- Tabelle: Bestellung
 - Id
 - Technischer Schlüssel (auto_increment bzw. Sequence)
 - fk_filiale
 - Fremd Schlüssel auf Tabelle Filiale
 - fk_kunde
 - Fremd Schlüssel auf Tabelle Kunde
 - fk_pizza
 - Fremd Schlüssel auf Tabelle Pizza
 - BestellDatum
 - Datum der Bestellung

1. Datenbank Design

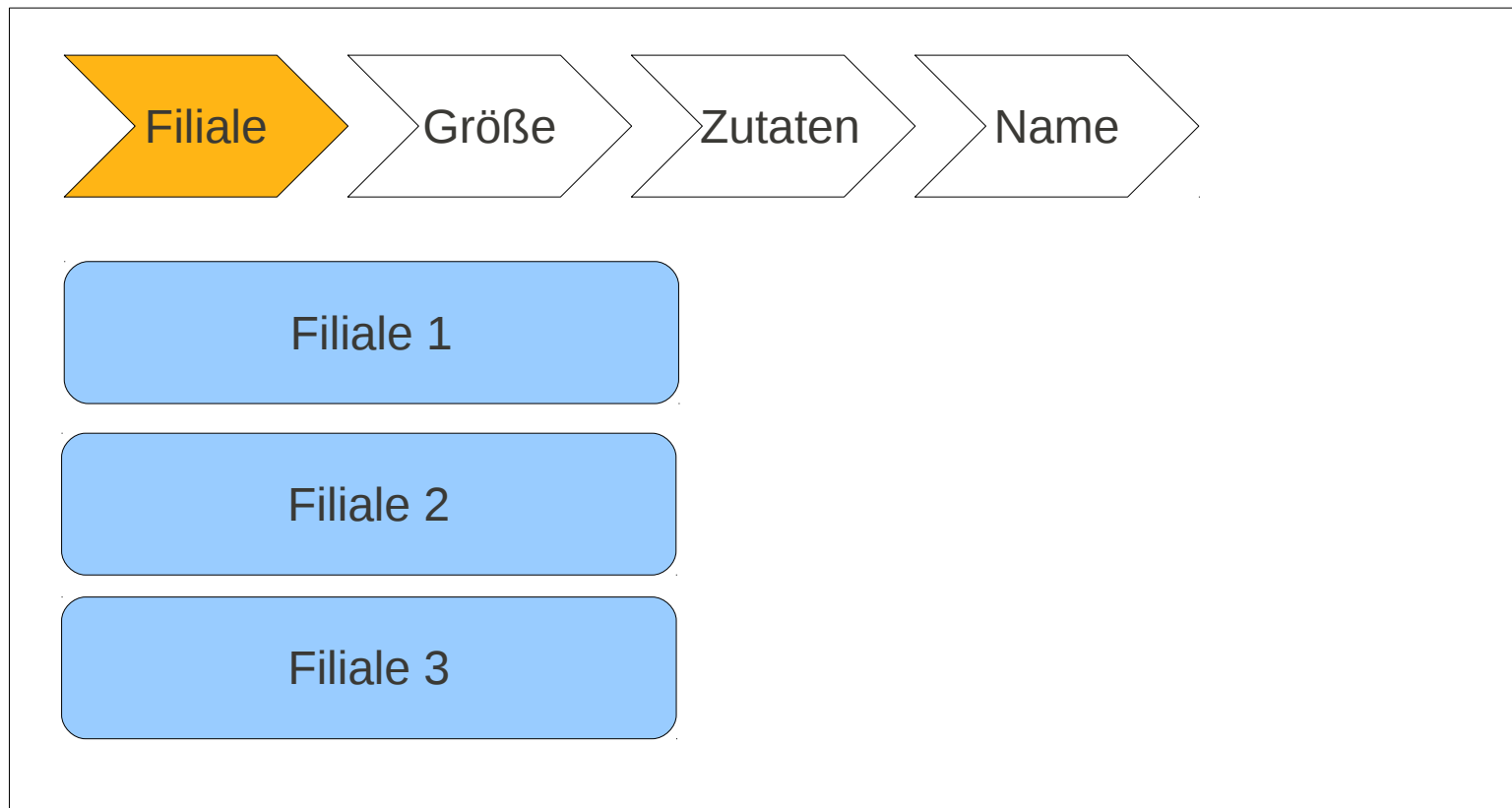
Fragen zu den Anforderungen

- Es bestehen gewisse Limitierungen basierend auf den Anforderungen:
 - Der Kunde kann nur eine Pizza innerhalb eines Bestellvorganges bestellen:
 - Ist das so gewünscht?
 - Zutaten sind in der vorliegenden Form nur in Form von Ja/Nein hinzu ffügbar:
 - Eventuell auch 2 oder 5 Scheiben Salami oder 50 gr. Käse?
 - Begrenzung der Attribute:
 - z.B. Name max. 30 Zeichen etc.?

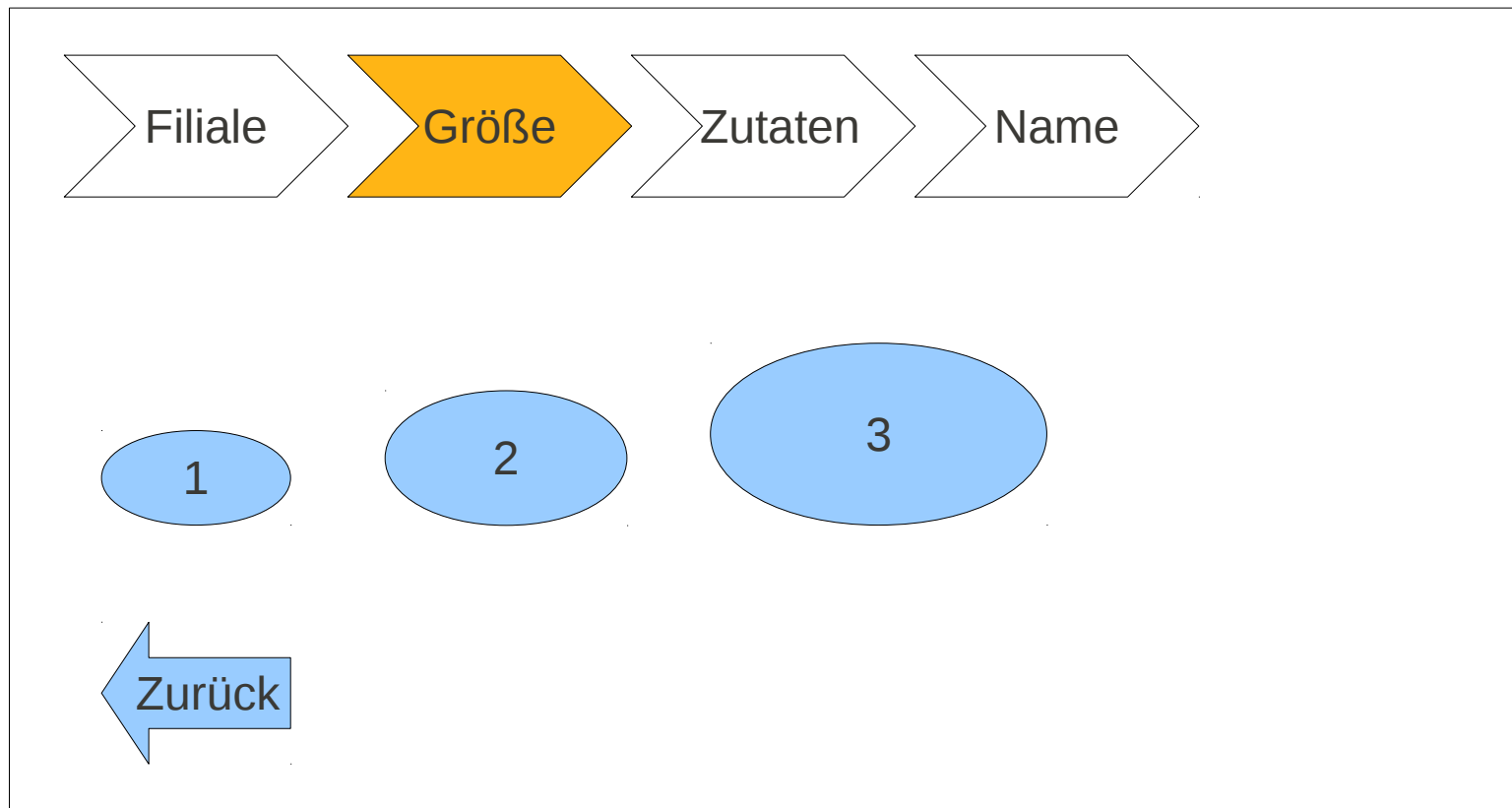
2. UML Diagramm Klassenhierarchie



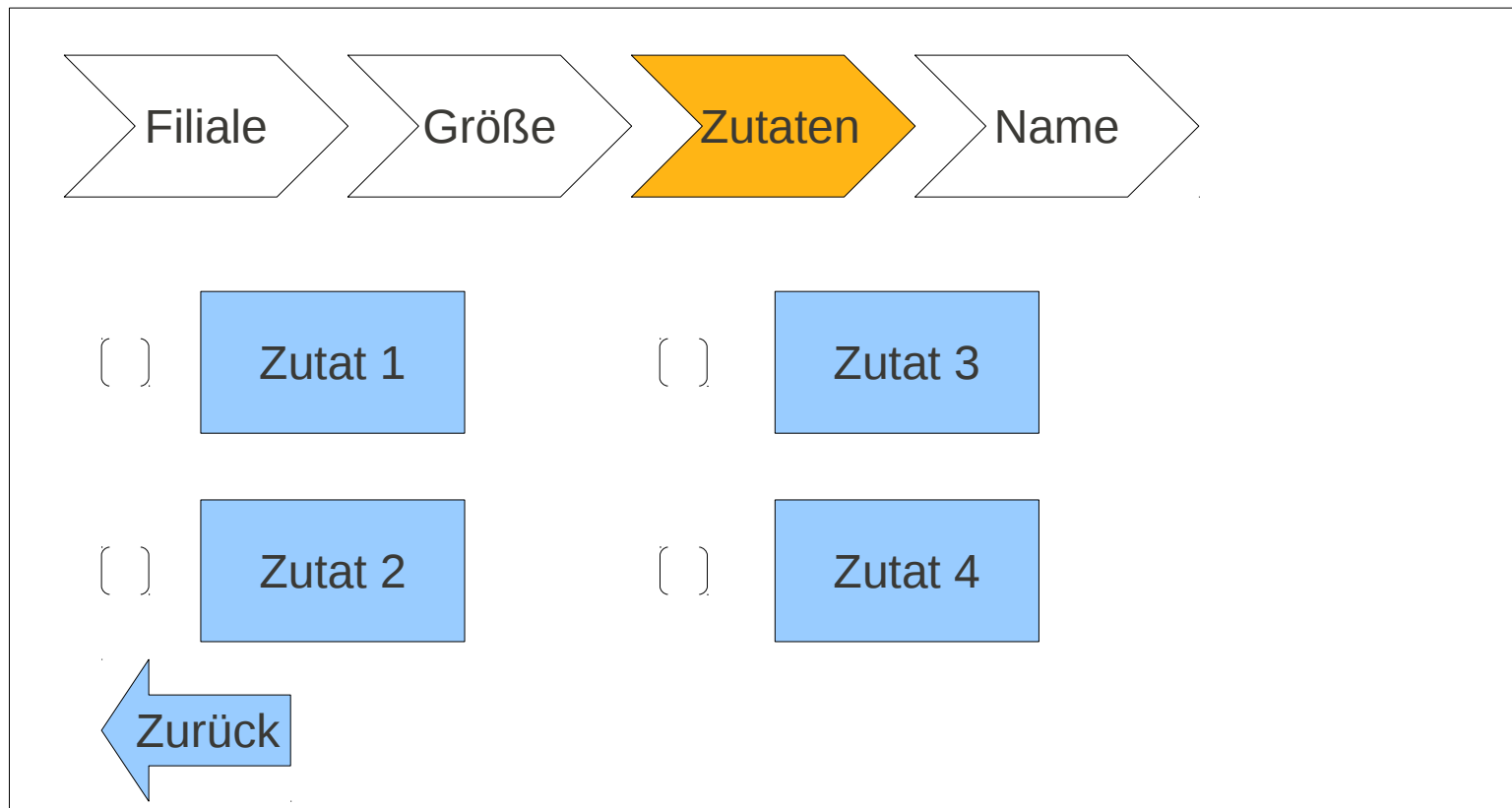
3. User Interface



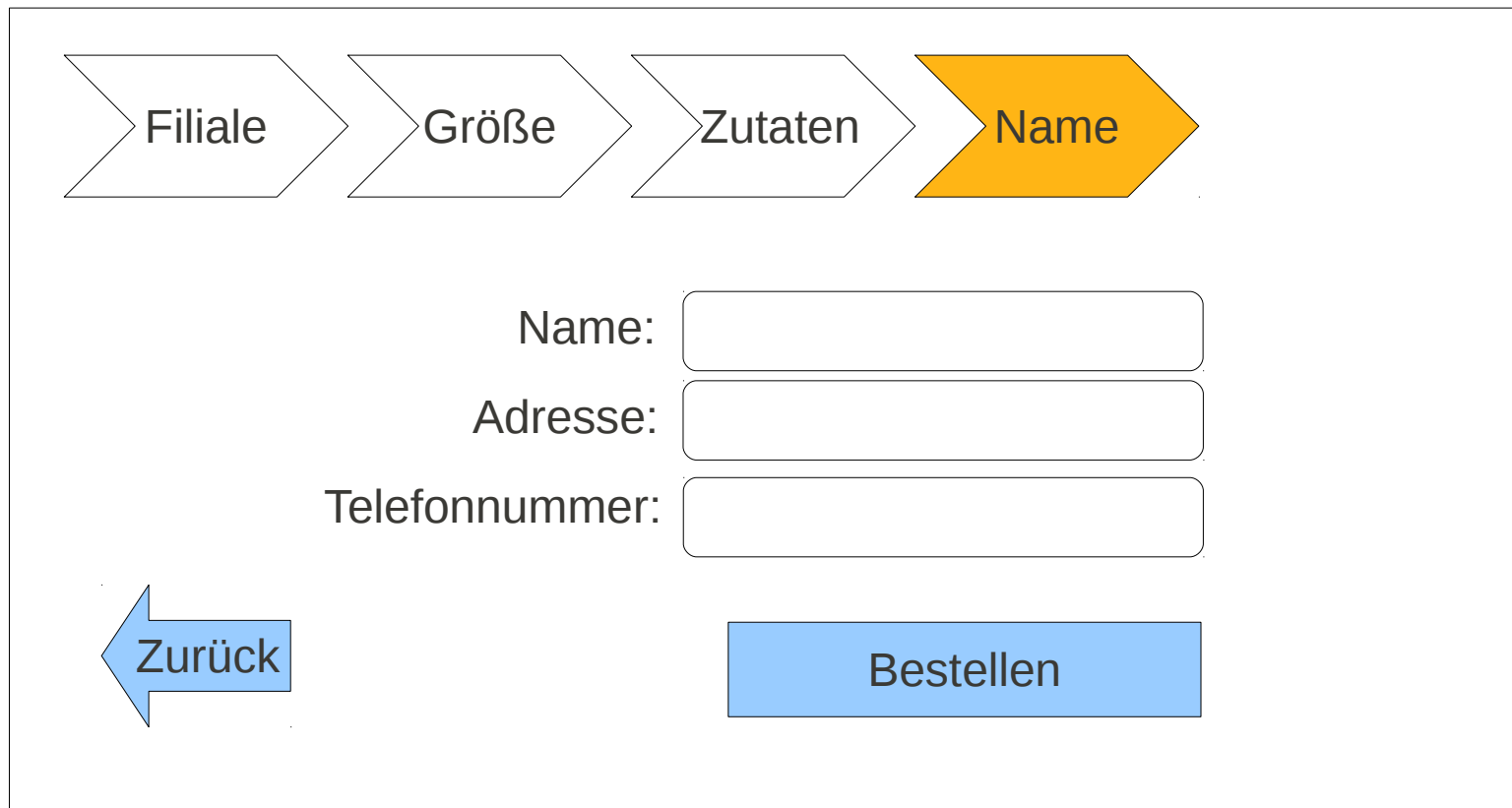
3. User Interface



3. User Interface



3. User Interface



The user interface is contained within a light gray rectangular frame. At the top, there is a horizontal sequence of four chevron-shaped buttons pointing to the right. The first three buttons are white with black outlines and contain the text 'Filiale', 'Größe', and 'Zutaten' respectively. The fourth button is solid orange and contains the text 'Name'. Below this sequence, on the left side, are three stacked text input fields. Each field is preceded by a label: 'Name:', 'Adresse:', and 'Telefonnummer:'. To the left of the bottom input field is a blue button with a white left-pointing chevron and the text 'Zurück'. To the right of the bottom input field is a solid blue rectangular button with the text 'Bestellen'.

Filiale Größe Zutaten Name

Name:

Adresse:

Telefonnummer:

Zurück Bestellen

4. SQL Abfrage

- 1. Schritt: Datumsbereich eingrenzen

```
SELECT
    *FROM
        bestellung
WHERE
    bestelldatum >= DATE_SUB(CURDATE(),INTERVAL 10 DAY);
```


4. SQL Abfrage

- 2. Schritt Anzahl Bestellungen

```
SELECT
    bestellung.*,
    count(*) AS AnzahlBestellungen
FROM
    bestellung
WHERE
    bestelldatum >= DATE_SUB(CURDATE(),INTERVAL 10 DAY)
HAVING AnzahlBestellungen >= 3;
```

4. SQL Abfrage

- 3. Schritt: Bestellungen mit Ausschluss der Zutaten

```
SELECT
    DISTINCT B.*
FROM
    bestellung AS B,
    zutat AS Z,
    zutaten_liste as ZL
WHERE
    B.fk_pizza = ZL.pizza_id
    AND Z.id = ZL.zutatenListe_ID
    AND NOT (Z.bezeichnung like 'Zutat1' OR Z.bezeichnung like 'Zutat2');
```

4. SQL Abfrage

- 4. Schritt: Die Kombination:

```
SELECT
    k.*,
    count(*) AS AnzahlBestellungen
FROM
    bestellung AS best
    LEFT JOIN kunde AS k USING(id)
WHERE
    best.bestelldatum >= DATE_SUB(CURDATE(),INTERVAL 10 DAY)
    AND best.id IN (
        SELECT DISTINCT B.ID
        FROM
            bestellung AS B,
            zutat AS Z,
            zutaten_liste as ZL
        WHERE
            B.fk_pizza = ZL.pizza_id
            AND Z.id = ZL.zutatenListe_ID
            AND NOT (Z.bezeichnung like 'Zutat1' OR Z.bezeichnung like 'Zutat2')
        )
    )
HAVING AnzahlBestellungen >= 1;
```