**Pizza-Datenbank**

*Adrian Locher & Leonard Schütz*

**Beschreibung:**

Wir haben uns dazu entschieden eine Datenbank für einen Pizzalieferdienst zu entwerfen.

Ein Pizza-Typ enthält ein bis mehrere Toppings. Zu den Toppings gehören z.B. Tomatensauce, Crème fraîche, Käse, Pilze, Salami oder sonstige Toppings. Pizzas können bestellt werden. Eine Bestellung kann mehrere Arten von Pizzas in unterschiedlichen Quantitäten enthalten. Zum Beispiel können zwei Margheritas, eine Salami und drei Fungi-Pizzas bestellt werden. Eine Bestellung wird von einem Kunden bestellt, und von einem Lieferboten geliefert. Pizza-Typen können immer wieder in Bestellungen vorkommen. Ein Kunde kann null bis viele Bestellungen aufgeben und ein Bote kann null bis viele Bestellungen austragen. Eine Bestellung wird immer von einem Kunden ausgelöst und von einem Boten ausgetragen.

**Konsistenzbedingungen:**

Der Preis eines Toppings liegt immer zwischen 2 und 10 Schweizer Franken.

Toppings können nicht doppelt auf einer Pizza vorkommen.

Es können nicht mehr als 2 extra Toppings und nicht mehr als 6 Toppings insgesamt auf einer Pizza bestellt werden.

Die Sauce gehört auch zu den Toppings, dafür hat die Pizza selbst keinen eignen Preis.

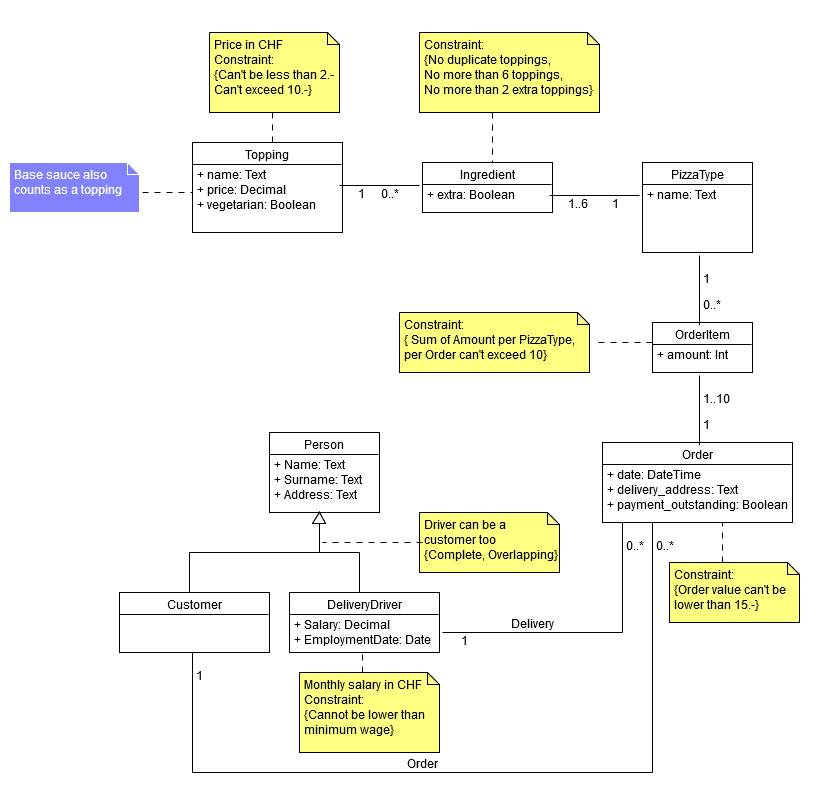
Pro Bestellung können mehrere Pizzas mit demselben Typ hinzugefügt werden, aber die Anzahl Pizzas pro Bestellung darf insgesamt nicht mehr als 10 sein. (Auch nicht von verschiedenen Pizza-Typen).

Der mindestbestellwert liegt bei 15 Franken.

Ein Lieferbote kann auch ein Kunde sein.

Der Lieferbote hat einen Lohn welcher nicht niedriger als der Mindestlohn sein darf.

**Domain-Modell:**



**Relationales Modell:**

1. person(
2. id INTEGER,
3. name VARCHAR(80) NOT NULL,
4. surname VARCHAR(80) NOT NULL,
5. address VARCHAR(80) NOT NULL,
6. )
7. customer(
8. id INTEGER,
9. person\_id INTEGER REFERENCES Person
10. )
11. delivery\_driver(
12. id INTEGER,
13. Person\_id INTEGER REFERENCES Person,
14. salary DECIMAL(10,2) NOT NULL,
15. employment\_date Date NOT NULL
16. )
17. order(
18. id INTEGER,
19. delivery\_driver INTEGER REFERENCES DeliveryDriver,
20. customer INTEGER REFERENCES Customer,
21. date DATE NOT NULL,
22. delivery\_address VARCHAR(80) NOT NULL,
23. payment\_outstanding BOOLEAN NOT NULL,
24. )
25. order\_item(
26. id INTEGER,
27. order INTEGER REFERENCES Order,
28. pizza\_type INTEGER REFERENCES PizzaType,
29. amount INTEGER
30. )
31. pizza\_type(
32. id INTEGER,
33. name VARCHAR(30) NOT NULL UNIQUE
34. )
35. ingredient(
36. id INTEGER,
37. pizza\_type INTEGER REFERENCES PizzaType,
38. topping INTEGER REFERENCES Topping,
39. extra BOOLEAN NOT NULL
40. )
41. topping(
42. id INTEGER,
43. name VARCHAR(30) NOT NULL UNIQUE,
44. price DECIMAL(4,2) NOT NULL,
45. vegetarian BOOLEAN NOT NULL
46. )

**Vererbung:**

Um die Vererbung abzubilden haben wir uns für eine Supertabelle mit je einer Untertabelle für Delivery-Driver und Customer entschieden. Da wir potenziell tausende Kunden abbilden, jedoch nur wenige Angestellte haben, würde nur eine Tabelle für Personen zu sehr vielen NULL Zellen führen. Dies wäre eine grosse Verschwendung. Die DeliveryDriver- und Customer-Tabelle enthält eine Reference auf die Person Tabelle.

Falls in Zukunft andere Berufsarten abgebildet werden sollten, kann man die DeliveryDriver Tabelle in Employees umbenennen und ein enum-Feld hinzufügen welche die Job-Art beschreibt.

Die Tabellen DeliveryDriver und Customer enthalten wenig Teststeinträge, da es sich dabei um Sub-Tabellen von Person handelt.