

Name:

Pizzageddon

Menü ansehen:

Akteur: Besteller

Ein Gast wird sich über das online Portal informieren welche Pizzen er sich bestellen kann und bei welchen es keine Zutaten mehr gibt. abhängig von seinem Standort wird angegeben wie lange die Lieferung dauern könnte, einfach ablesbar ist auch Preis und Zutaten.

Pizza bestellen:

Akteur: Besteller

Der Gast hat sich eine oder mehrere Pizzen zum Bestell ausgesucht und hat den Kauf bestätigt, die kaufentscheidung wird an die Datenbank weitergeleitet und der Käufer bekommt eine Angabe wie lange das Backen dauern könnte.

Bestellung erfassen:

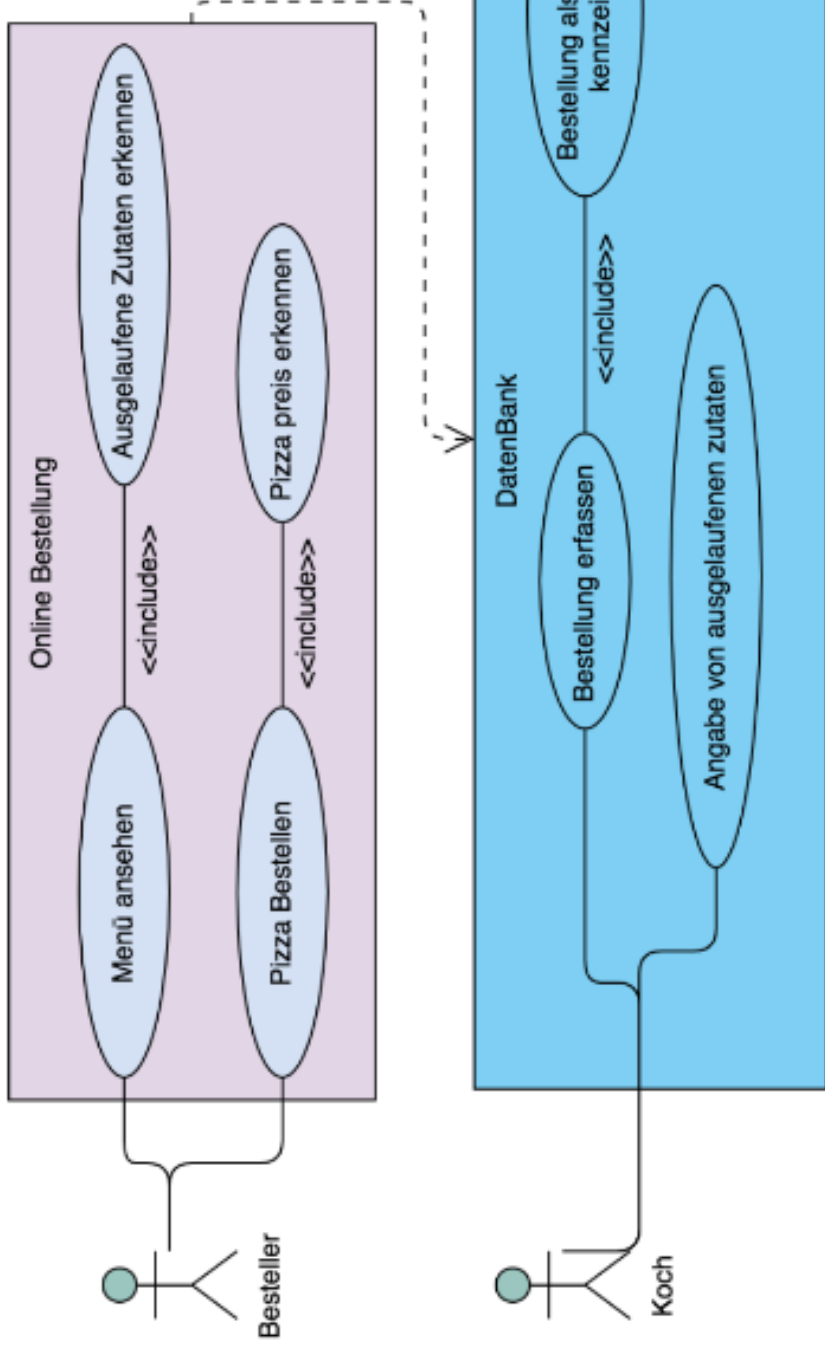
Akteur: Koch

da nun eine bestellung in der datenbank vorliegt kann der koch nun das backen starten, zusätzlich kennzeichnet er die bestellung als bearbeitet.

Angabe zu abgelaufenen Zutaten:

Akteur: Koch

Um den besteller aufzuklären welche Pizzen gemacht werden können kann der Koch mit der Datenbank angeben welche Pizzen machbar sind.



Besteller: id, bestellung, adresse

{Primary Key: id}

{Unique: bestellung, adresse}

Koch: id, Cheese, Tomato, ..., Lieferdauer

{Primary Key: id}

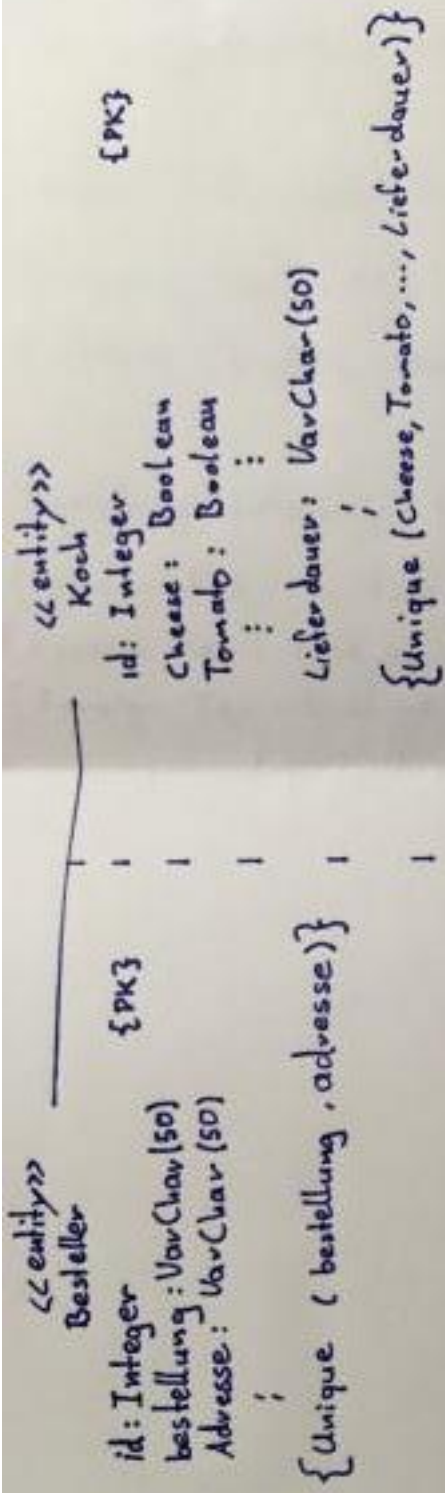
{Unique: Cheese, Tomato, ..., Lieferdauer}

bestellung / Lieferung: b-id, k-id, Lieferung

{Primary Key: b-id, k-id}

{Foreign Key: b-id \rightarrow Besteller: id}

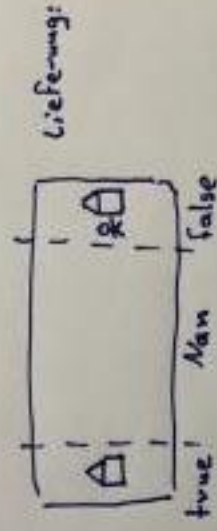
{Foreign Key: k-id \rightarrow Koch: id}



<<relationship>>
 Bestellung/Lieferung

b_id: Integer {PK, FK}
 k_id: Integer {PK, FK}

Lieferung: Boolean



```

CREATE TABLE besteller
(b_id    INTEGER    NOT NULL,
 b_bestellung  VARCHAR(30) NOT NULL,
 b_adresse VARCHAR(50) NOT NULL,

CONSTRAINT besteller
PRIMARY KEY (b_id),

CONSTRAINT unique_besteller_bestellung_adresse
UNIQUE (b_bestellung, b_adresse)
);

```

```

CREATE TABLE koch
(k_id    INTEGER    NOT NULL,
 k_cheese Boolean
 k_tomato Boolean
 k_lieferdauer VARCHAR(50) NOT NULL,

CONSTRAINT koch
PRIMARY KEY (k_id),

CONSTRAINT unique_koch_cheese_tomato_lieferdauer
UNIQUE (k_cheese, k_tomato, k_lieferdauer)
);

```

```

CREATE TABLE lieferung
(b_id    INTEGER    NOT NULL,
 k_id    INTEGER    NOT NULL,
 l_lieferung: Boolean NOT NULL,

CONSTRAINT pk_lieferung
PRIMARY KEY (b_id, k_id, l_lieferung),

CONSTRAINT fk_lieferung_besteller
FOREIGN KEY (b_id) REFERENCES besteller (b_id),

CONSTRAINT fk_lieferung_koch
FOREIGN KEY (k_id) REFERENCES koch (k_id),
);

```

```

INSERT INTO besteller(b_id, b_bestellung, b_adresse)
VALUES
(1, 'käse pizza' , 'ziegeleistr.14'),
(2, 'Tomaten pizza' , 'irgendeinestr.7'),
;

```

```

INSERT INTO koch(k_id, k_cheese, k_tomato, k_lieferdauer)
VALUES
(1, 'True', 'True', '2'),
(2, 'True', 'True', '1'),
;

```

```

INSERT INTO liefert(b_id, k_id, l_lieferung)
VALUES
( 1, 1, Null)
( 1, 2, Null)
( 2, 1, Null)

```