

**BERUFLICHE SCHULE ITECH**  
Elbinsel Wilhelmsburg

## PROJEKT 01

DOKUMENTATION

# Marktpersonal

LAGER

*Christian Rudolph, Artem Zheleznyakov, Felix Schulz, Patrick Rau*



IDK-COMPANY

27. Juni 2018

# Inhaltsverzeichnis

## 1 Vorbereitung

## 2 Kundengespräch

- 2.1 Kundenwünsche . . . . .
- 2.2 Resultierendes Angebot . . . . .

## 3 Datenbank zur Geräteverwaltung

- 3.1 Generierung der Datenbank . . . . .
- 3.2 Datenbankabfragen . . . . .

## 4 Anhang / Abbildungen

# Abbildungsverzeichnis

- 1 ERD zur Datenbank . . . . .
- 2 SQL-Statement zum Generieren der Datenbank . . . . .
- 3 Ausgabe aller Komponenten mit Leistungsmerkmalen sortiert nach Art und dann nach Preis . . . . .
- 4 Ausgabe aller Komp. ohne Ergonomiemerkmale . . . . .
- 5 Ausgabe der teuersten, preiswertesten und durchschnittlichen Preises für einen Komponententyp . . . . .
- 6 Gesamtübersicht aller Gütesiegel . . . . .
- 7 Einzelpreis ändern . . . . .
- 8 Tablet-location Suche . . . . .
- 9 Alle Produkte eines Herstellers . . . . .
- 10 Alle Produkte einer Komponentenart . . . . .
- 11 Produkte mit best. Siegel . . . . .
- 12 Anzahl des Bestandes aller Komponenten . . . . .

## **Zusammenfassung**

Im Folgenden wird dem Vorgang der Erstellung eines Angebotes beschrieben. Der Fokus liegt hierbei auf dem Einsatz eines IT-Systems im Logistikbereiches der Baufuchs GmbH. Dabei beschäftigt sich dieser Artikel mit der Implementierung einer Datenbank, der Erstellung von Angeboten, einem Kundengespräch wie auch die Erstellung eines englisch- und deutschsprachigen Informationsblattes.

## **1 Vorbereitung**

Um die BAUFUCHS GMBH als Kunden für die Lösung zu gewinnen, muss in erster Linie ein Kundengespräch vorbereitet werden. Aufgrund der Mobilität unser eingesetzten Tablets bietet es sich an diese in der Logistik zu verwenden. Weitere Vorteile dieser Geräte werden im Laufe der Dokumentation genauer beschrieben.

Für die Implementierung dieses IT-Systems wird zusätzlich eine Datenbank angelegt um das Lagerhaus optimal zu verwalten und es soweit anzupassen, dass es möglich wäre dieses System mit dem Kassensystem zu verknüpfen.

Da der Gesundheitsschutz der BAUFUCHS GMBH eine hohe Priorität zukommt, wird für diese Zwecke ein Flyer erstellt um die Mitarbeiter zu informieren. Aufgrund der Internationalität der BAUFUCHS GMBH wird dieser in englischer wie auch deutscher Sprache angeboten.

Optional wird eine Software zur Visualisierung als Demo angeboten.

## **2 Kundengespräch**

Um eine optimale Lösungsfindung zu garantieren werden zwei Kundengespräche durchgeführt. Im ersten Gespräch werden die Kundenwünsche aufgenommen und im Zweiten werden Angebote präsentiert und diese diskutiert. Aus dem Ergebnis beider resultiert das für den Kunden zugeschnittene Angebot.

### **2.1 Kundenwünsche**

Das Lager befindet sich im Geschäft auf den Regalen. Dementsprechend kommt es daher zu besonderen Prioritäten.

Aus dem Gespräch konnte die hohe Priorisierung der Akkulaufzeit, wie auch eine erhöhte Priorität des Tabletdesigns festgestellt werden. Das Design ist für den Kunden in sofern relevant, da es durchaus das Unternehmen widerspiegelt. Durch den besonderen Aufbau des Geschäfts ist es nötig, dass das Betriebssystem von jedem Kunden ohne weitere Probleme bedient werden kann. Die Kamera ist in sofern wichtig, da QR-Codes ohne Probleme eingescannt werden müssen. Kosten sind für den Kunden nicht relevant. Aus dem Gespräch resultierte ebenfalls, dass kein weiteres Gespräch vonnöten sein wird.

### **2.2 Resultierendes Angebot**

Aus den Kundenwünschen konnten drei Angebote erstellt werden.

## **3 Datenbank zur Geräteverwaltung**

In diesem Teil wird die Implementierung der Geräte Datenbank behandelt. Hinzu kommen die entsprechenden SQL-Statements die benötigt werden um die entsprechende Datenbank zu erstellen, wie zudem die entsprechenden SELECT Statements um Daten aus der Datenbank abzufragen. Am Ende dieses Abschnitts werden außerdem die Probleme, die beim Erstellen der Statements aufgetreten sind, genauer erläutert.

### 3.1 Generierung der Datenbank

Um unsere Datenbank zu erstellen wurde ein ERD entworfen und verwendet um die Datenbank zu generieren. Siehe Abbildung 1 Nun wird der Aufbau der Datenbank beschrieben, welche aus dem SQL-Statement (siehe Abbildung 2). In Zeile 1 erstellen wir eine Datenbank mit dem Namen `DEVICE_MANAGEMENT`. Nun werden mehrere Tabellen erstellt die wir benutzen wollen (siehe Zeilen 5, 9, 15, 19, 30, 36, 40, 47). Als Tabellen wurden *component\_type*, *seal*, *manufacturer*, *product\_designation*, *special\_requirements*, *product\_seal* und *product\_designation* gewählt.

Die Entscheidungen der Datentypen sind größtenteils trivial, jedoch wurde die Entscheidung getroffen den Preis in Cent anzugeben. Der Grund für die Entscheidung den Preis in Cent anzugeben ist, dass es somit weit einfacher und fehlerfreier ist Berechnungen durchzuführen. Double wurde nicht gewählt, da es zu Präzisionsproblemen kommen kann die bereits bei elementaren Rechnungen auftreten könnten.

### 3.2 Datenbankabfragen

## 4 Anhang / Abbildungen

Abbildung 1: ERD zur Datenbank

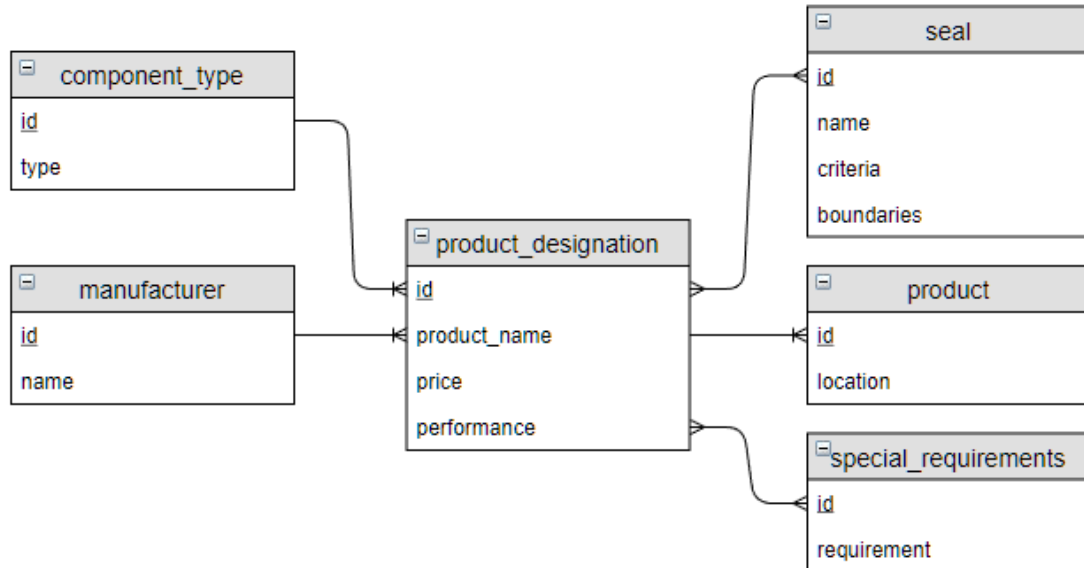


Abbildung 2: SQL-Statement zum Generieren der Datenbank

```
1 CREATE DATABASE IF NOT EXISTS device_management;
2
3 USE device_management;
4
5 CREATE TABLE IF NOT EXISTS component_type(
6 id INTEGER NOT NULL PRIMARY KEY AUTO_INCREMENT,
7 type VARCHAR(255));
8
9 CREATE TABLE IF NOT EXISTS seal(
10 id INTEGER NOT NULL PRIMARY KEY AUTO_INCREMENT,
11 name VARCHAR(255),
12 criteria VARCHAR(255),
13 boundaries VARCHAR(255));
14
15 CREATE TABLE IF NOT EXISTS manufacturer(
16 id INTEGER NOT NULL PRIMARY KEY AUTO_INCREMENT,
17 name VARCHAR(255));
18
19 CREATE TABLE IF NOT EXISTS product_designation(
20 id INTEGER NOT NULL PRIMARY KEY AUTO_INCREMENT,
21 product_name VARCHAR(255),
22 -- price in cent
23 price INTEGER,
24 performance VARCHAR(255),
25 type_id INTEGER,
26 manufacturer_id INTEGER,
27 FOREIGN KEY(type_id) REFERENCES component_type(id),
28 FOREIGN KEY(manufacturer_id) REFERENCES manufacturer(id));
29
30 CREATE TABLE IF NOT EXISTS product(
31 id INTEGER NOT NULL PRIMARY KEY AUTO_INCREMENT,
32 location VARCHAR(255),
33 designation_id INTEGER,
34 FOREIGN KEY(designation_id) REFERENCES product_designation(id));
35
36 CREATE TABLE IF NOT EXISTS special_requirements(
37 id INTEGER NOT NULL PRIMARY KEY AUTO_INCREMENT,
38 requirement VARCHAR(255));
39
40 CREATE TABLE IF NOT EXISTS product_seal(
41 product_id INTEGER NOT NULL,
42 seal_id INTEGER NOT NULL,
43 FOREIGN KEY(product_id) REFERENCES product_designation(id),
44 FOREIGN KEY(seal_id) REFERENCES seal(id),
45 PRIMARY KEY(product_id, seal_id));
46
47 CREATE TABLE IF NOT EXISTS product_requirements(
48 product_id INTEGER NOT NULL,
49 requirement_id INTEGER NOT NULL,
50 FOREIGN KEY(product_id) REFERENCES product_designation(id),
51 FOREIGN KEY(requirement_id) REFERENCES special_requirements(id),
52 PRIMARY KEY(product_id, requirement_id));
```

Abbildung 3: Ausgabe aller Komponenten mit Leistungsmerkmalen sortiert nach Art und dann nach Preis

```
1 SELECT product_name, performance, price FROM product_designation
2 ORDER BY type_id, price;
```

Abbildung 4: Ausgabe aller Komp. ohne Ergonomiemerkmale

```
1 SELECT product_name, performance, price FROM product_designation AS
   p
2 LEFT JOIN product_requirement AS pr ON p.id = pr.product_id
3 WHERE pr.product_id IS NULL;
```

Abbildung 5: Ausgabe der teuersten, preiswertesten und durchschnittlichen Preises für einen Komponententyp

```
1 SELECT product_name, price FROM product_designation
2 WHERE type_id = (SELECT id FROM component_type WHERE type = [
   component_type])
3 ORDER BY price DESC
4 LIMIT 1 ;
5
6 SELECT product_name, price FROM product_designation
7 WHERE type_id = (SELECT id FROM component_type WHERE type = [
   component_type])
8 ORDER BY price ASC
9 LIMIT 1 ;
10
11 SELECT AVG(price) FROM product_designation
12 WHERE type_id = (SELECT id FROM component_type WHERE type = [
   component_type]);
```

Abbildung 6: Gesamtübersicht aller Gütesiegel

```
1 SELECT name, criteria, boundaries FROM seal;
```

Abbildung 7: Einzelpreis ändern

```
1 UPDATE product_designation
2 SET price = [new_price]
3 WHERE id = [product_id];
```

Abbildung 8: Tablet-location Suche

```
1 SELECT location FROM product
2 WHERE id = [product_id];
```

Abbildung 9: Alle Produkte eines Herstellers

```
1 SELECT product_name FROM product_designation
2 WHERE type_id = (SELECT id FROM component_type WHERE name = [
   manufacturer_name]);
```

Abbildung 10: Alle Produkte einer Komponentenart

```
1 SELECT product_name FROM product_designation
2 WHERE manufacturer_id =(SELECT id FROM manufacturer WHERE name = [
    component_type]);
```

Abbildung 11: Produkte mit best. Siegel

```
1 SELECT product_name FROM product_designation
2 WHERE seal_id =(SELECT id FROM seal WHERE name = [seal_name]);
```

Abbildung 12: Anzahl des Bestandes aller Komponenten

```
1 SELECT name, count(name) FROM product
2 GROUP BY name;
```