# Dokumentation Projekt M153 Datenbank

## Aeschliman Lian & Odermatt Joshua

## Beschreibung

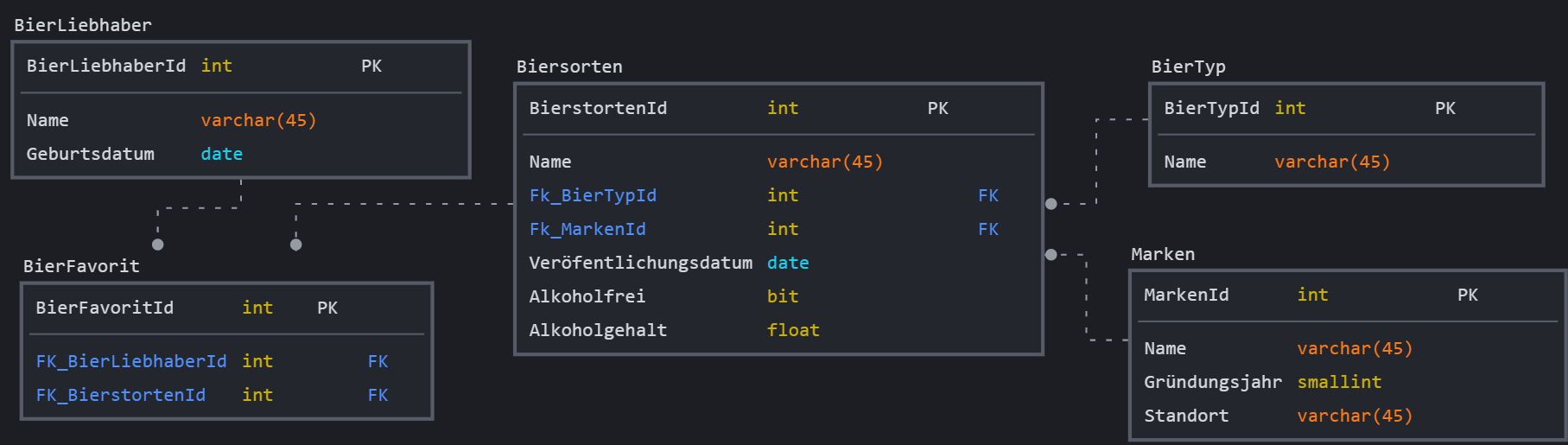
In unserer Umsetzung haben wir eine Datenbank für Bierliebhaber gemacht. So zusagen ein Bierlexikon.

In der Datenbank haben wir Tabellen zur Anlegung von verschieden Biermarken, Biertypen, Bier-Produkte und natürlich Bier Liebhaber umgesetzt. Jeder Bier Liebhaber hat bzw. kann verschiedene Biere als Favoriten speichern.

Die Umsetzung haben wir in MYSQL gemacht, da MYSQL sehr einfach mit Docker auch auf Linux sowie Mac’s virtualisierbar ist und wir somit wenig Zeit mit dem Setup verbringen mussten.

In dieser Dokumentation ist das ER-Diagramm, eine detaillierte Beschreibung zu allen Tabellen, ein Beschrieb der Manipulationen und Abfragen, ein Beschrieb der umgesetzten Trigger und Stored-Procedures sowie der SQL-Code zu finden.

## ER-Diagramm



## Detail Beschrieb

### BierLiebhaber

BierLiebhaberId int PK = Primärschlüssel der Tabelle BierLiebhaber

Name varchar(45) = Name des Bier-Liebhabers

Vorname varchar(45) = Vorname des Bier-Liebhabers

Geburtsdatum date = Geburtsdatum des Bier-Liebhabers

### BierFavorit

BierFavoritId int PK = Primärschlüssel der Tabelle BierFavorit

FK\_BierLiebhaberId = Fremdschlüssel zur Tabelle BierLiebhaber

FK\_BiersortenId = Fremdschlüssel zur Tabelle Biersorten

### Biersorten

BIersortenId int PK = Primärschlüssel der Tabelle Biersorten

Name varchar(45) = Name der Biersorte

FK\_BierTypId = Fremdschlüssel zur Tabelle BierTyp

FK\_MarkenId = Fremdschlüssel zur Tabelle Marken

Veröffentlichungsjahr int = Einführungsjahr der Biersorte

Alkoholfrei bit = Ist die Sorte Alkoholfrei(true/1) oder nicht(false/0)

Alkoholgehalt float = Alkoholgehalt der Sorte in %

### BierTyp

BierTypId int PK = Primärschlüssel der Tabelle BierTyp

Name varchar(45) = Name des Bier-Typs

### Marken

MarkenId int PK = Primärschlüssel der Tabelle Marke

Name varchar(45) = Name der Marke

Gründungsjahr smallint = Gründungsjahr der marke

Standort varchar(45) = Standort der Biermarke

## Trigger und Stored-Prcedures

### Trigger

#### gruendungsjahr\_ueberprüfen

Der Umgesetzte Trigger soll sicherstellen, ob bei der Tabelle Marke kein fälschliches Gründungsjahr eingetragen wird. Der Trigger überprüft, ob es sich beim Gründungsjahr um ein vergangenes oder das Aktuelle Jahr handelt. Liegt das Jahr in der Zukunft, so wird ein Error geworfen und das einfügen des Datensatzes wird abgebrochen. Der Trigger deklariert zwei Variablen. Zum einen die Error-Message und zum anderen das Aktuelle Jahr. Das Aktuelle Jahr wird als int gespeichert und den Funktionen YEAR(NOW()) geholt. Danach wird verglichen, ob es das aktuelle Jahr < das eingetragen Jahr ist. Ist dem so, so wird der Eintrag verhindert.

### Stored-Procedures

#### sp\_DurchschnitAlkoholgehaltVon

Diese Prozedur kann gebraucht werden um den durchschnittlichen Alkoholgehalt von einer bestimmten Marke heraus zu finden.

Am Beginn der Prozedur wird überprüft ob die Marke existiert, wenn dies nicht der Fall ist wird ein Error geworfen. Falls die Marke existiert wird der durchschnittliche Alkoholgehalt zurückgegeben. Der Durchschnitt wird mit der AVG() Methode in einer simplen SELECT Abfrage berechnet.

#### get\_liebhaber\_amount\_of\_beer

Die erste umgesetzte Procedure dient dazu die Anzahl der Leute bzw. BierLiebhaber zu ermitteln, welche eine bestimmte Biersorte als Favoriten gespeichert haben. Dazu wird der Name der Sorte als Parameter übergeben. Danach wird mit COUNT() und zwei joins auf die Tabellen BierFavoriten und BierSorten die Anzahl ermittelt.

#### delete\_products\_of\_marke

Die zweite Procedure dient dazu, die verwandten Daten einer Marke zu löschen. Sozusagen ein cascade Verfahren. Übergeben wird dazu die Marken ID. Bei den verwandten Daten einer Marke handelt es sich um die zur Marke gehörenden Bier Sorten sowie die Favoriten, welche wiederum die Bier Sorten referenzieren. Das Procedure verwendet ein Subquery um die IDs der Bier Sorten, welche zu einer Marke gehören, zu ermitteln. Danach werden alle Favoriten, welche Fremdschlüssel zu einer dieser Sorten haben gelöscht. Auch werden danach alle Sorten gelöscht, welche mit der übergebenen Marken ID übereinstimmen.

## SQL-Code