# Dokumentation Datenbank Projekt

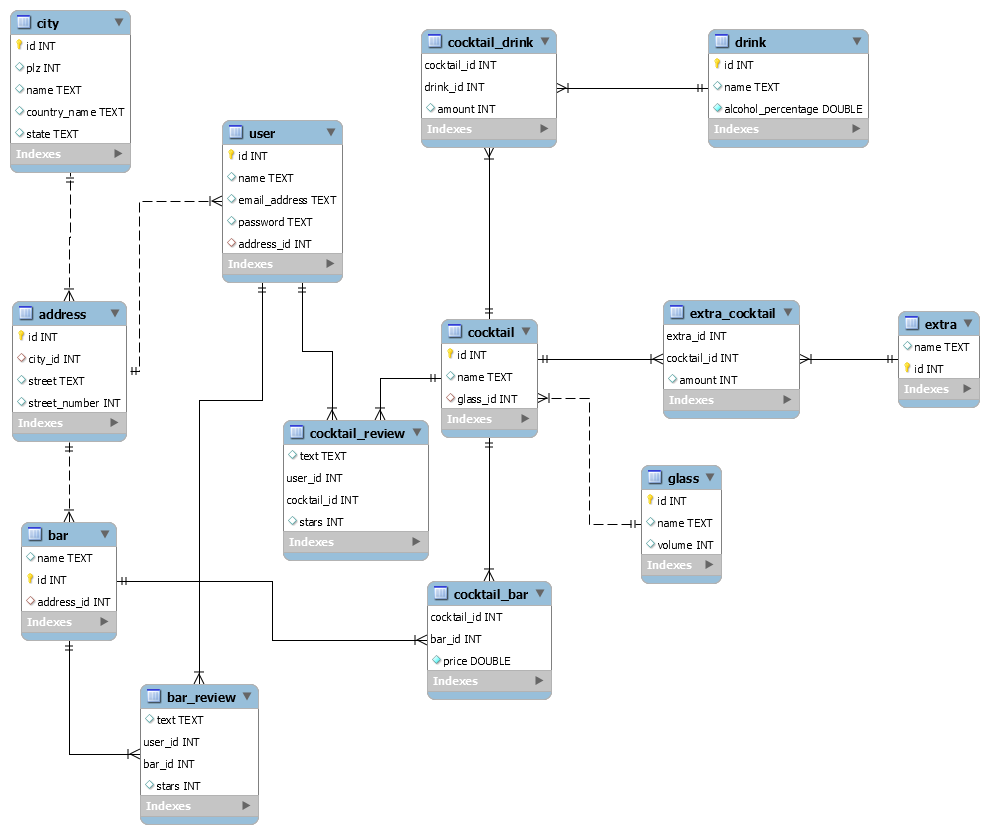
## Dokumentation inkl. Datenbankschema (Mindestanforderungen mehrere 1:n Beziehungen, 2\* m:n Beziehung, Kardinalität)

### Beschreibung der Idee: Für welchen Anwendungsfall wurde die Datenbank erstellt? Was soll alles darin gespeichert werden? Welche Art von Applikation nutzt diese Datenbank? Typische Anwendungsfälle beschreiben aus denen die Struktur der Datenbank klar wird. (Beispiele wie ein User auf die Daten der Datenbank zugreift/die Applikation nutzt für die die Datenbank erstellt wurde.)

Unsere Datenbank hält die Daten von Cocktails. Dabei werden auch Bars gespeichert.

Die Cocktail-Datenbank könnte von einer Webseite genutzt auf welcher man sich Cocktailrezepte anschauen kann und auch bewerten kann. Außerdem kann man auch schauen welche Bars welche Cocktails haben. **Mit der Webseite kann man den tollsten Cocktail-Abend planen**.

### Datenbankschema



#### Tabellen:

##### city

Die city Tabelle speichert die einzelnen Städte wie (Mainz, Wiesbaden, Berlin). Die Tabelle wird benötigt um keine Redundanzen zu haben. Ohne die city Tabelle würde für jede Stadt, das Land, das Bundesland und die PLZ gespeichert werden.

Da wir eine city Tabelle haben, speichern wir für jede Stadt nur einmal Land, PLZ, Bundesland.

In unserer Datenbank werden Stadtteile weggelassen. Stadtteile würden auch in der city tabelle gespeichert werden.

Obwohl man das Datenbankmodell natürlich auch noch mit einer Stadtteil Tabelle erweitern könnte.

##### address

Die Adresse speichert die Anschrift. Die Adresse hat eine Stadt, Straße und Hausnummer.

##### user

Der user ist ein möglicher Besucher unserer Webseite. Mit dem passwort kann er sich in unserer Seite einloggen.

##### bar

Die Bar hat einen Namen und eine Adresse.

##### bar\_review

Speichert den Text der Bewertung, die Anzahl Sterne (1-5), sowie die Bar für die die Bewertung geschrieben und den User der die Bewertung geschrieben hat

##### cocktail\_review

Sowie barreview nur für cocktails.

##### cocktail\_bar

Cocktail bar speichert welche Cocktails eine Bar anbietet und den Preis

##### cocktail

Cocktail speichert den cocktail namen und das Glas des Cocktails

##### glass

Glas hat einen Namen und ein Füllvermögen

##### drink

Ein Getränk welches in den Cocktail gemischt wird

##### cocktail\_drink

Ist die Beziehungstabelle zwischen cocktail und drink, speichert die Zutaten des Cocktails

##### extra

Speichert extras dazu gehören Eiswürfel und Zitronenscheiben, welche zu einem guten Cocktail dazu gehören

##### extra\_cocktail

Beziehungstabelle zwischen extra und cocktail.

Unsere Tabellen sind alle klein geschrieben. Die Attribute ebenso.

Zum trennen von Wörtern wurden \_ verwendet.

Naming Conventions SQL:

<https://dzone.com/articles/a-guide-to-sql-naming-conventions>

## Typische Fragestellungen an die Datenbank inkl. Select Statement (als Musterlösung): Mindestanforderung: Aufgaben in allen Schwierigkeitsgraden (siehe Kursarbeit):

### einfaches select

Gib alle Bars aus.

|  |
| --- |
| SELECT \* FROM bar |

### 

|  |  |  |
| --- | --- | --- |
| name | id | address\_id |
| Die GMO-Bar | 1 | 12 |
| Saufbar | 2 | 6 |
| FurchtBAR | 3 | 5 |

### 

### Where

Getränke mit höchstens 40% Gehalt.

|  |
| --- |
| SELECT \* FROM drink  WHERE drink.percentage <= 40 |

|  |  |  |
| --- | --- | --- |
| id | name | alcohol\_percentage |
| 1 | Wodka | 40 |
| 2 | Rum, weiß | 40 |
| 4 | Jägermeister | 35 |
| 5 | Vermouth | 22 |
| 6 | Gin | 37.5 |
| 7 | Bitburger | 4.8 |
| 8 | Energy Drink | 0 |
| 9 | Limettensaft | 0 |
| 10 | Zitronensaft | 0 |
| 11 | Cola | 0 |

### inner join

Anschriften in Mainz

|  |
| --- |
| SELECT \* FROM address INNER JOIN city  ON address.city\_id = city.id  WHERE city.name = 'Mainz' |

|  |  |  |
| --- | --- | --- |
| plz | street | street\_number |
| 55116 | Hechtsheimer-Straße | 12 |
| 55116 | Mainzer-Straße | 28 |
| 55116 | Große-Bleiche | 17 |
| 55116 | Hechtsheimer-Straße | 29 |

### inner join über mehrere Tabellen

Cocktails mit energy drink als drink

|  |
| --- |
| SELECT cocktail.name  FROM cocktail INNER JOIN cocktail\_drink INNER JOIN drink  ON cocktail.id = cocktail\_drink.cocktail\_id AND cocktail\_drink.drink\_id = drink.id  WHERE drink.name = 'Energy Drink' |

|  |
| --- |
| name |
| Wodka E |
| Jägerbomb |

### group by

Durchschnittliche Bewertung der Bars

|  |
| --- |
| SELECT bar.name , AVG(bar\_review.stars) AS Anzahl  FROM bar\_review INNER JOIN bar  ON bar\_review.bar\_id = bar.id  GROUP BY bar\_review.bar\_id |

|  |  |
| --- | --- |
| name | Anzahl |
| Die GMO-Bar | 4 |
| Saufbar | 3.3333 |
| FurchtBAR | 2.6667 |

### having

Bars mit mindestens 3 Sternen im Durchschnitt

|  |
| --- |
| SELECT bar.name , AVG(bar\_review.stars) AS Anzahl  FROM bar\_review INNER JOIN bar  ON bar\_review.bar\_id = bar.id  GROUP BY bar\_review.bar\_id  HAVING Anzahl > 3 |

|  |  |
| --- | --- |
| name | Anzahl |
| Die GMO-Bar | 4 |
| Saufbar | 3.3333 |

## Umsetzung in MySQL:

### Tabellen in der Datenbank erstellen inkl. Testdaten

### Separate Text Datei (.txt) mit: SQL Code der Datenbank (Tabellen selektieren => rechtklick => Copy to clipboard => Create Statement)

Die Datei

<https://docs.google.com/document/d/1upaNS7NvnTw-qHYZ_WwgenwjTCUaVM_n7b8cC8funiU/edit?usp=sharing>

### Separate Text Datei (.txt) mit: Kompletter Export der Datenbank als Create und Insert Statement (TeamX selektieren => Server => Data Export)

Die Datei

<https://docs.google.com/document/d/1SkvoPQ9wYuMsB80taiG0EqHygDZAngS8BnFX18IHajU/edit?usp=sharing>

## Angabe zu jedem Gruppenmitglied wieviel Prozent des Gruppenergebnisses auf die einzelnen Mitglieder zurückzuführen sind.

**Tom: 33,3%**

**Tobias: 33,3%**

**Jonas: 33,3%**

## Präsentation

<https://docs.google.com/presentation/d/1Gs5-0QBjUU7aBpaRJy53s9MBGl9AouXN23g10VtT25Y/edit?usp=sharing>