Datenbankprojekt – Quartett Spiel

1. Inhaltsverzeichnis

[1. Datenbank Beschreibung 2](#__RefHeading___Toc502_1161155005)

[2 ER-Modell 3](#__RefHeading___Toc552_4081095476)

[3 Relationsschreibweise 4](#__RefHeading___Toc554_4081095476)

[4 SQL-Statement 6](#__RefHeading___Toc556_4081095476)

[4.1 Eine Tabelle I 6](#__RefHeading___Toc558_4081095476)

[4.2 Eine Tabelle II 6](#__RefHeading___Toc560_4081095476)

[4.3 Eine Tabelle III 6](#__RefHeading___Toc562_4081095476)

[4.4 Mehrere Tabellen Ohne Inner Join I 7](#__RefHeading___Toc564_4081095476)

[4.5 Mehrere Tabellen Ohne Inner Join II 7](#__RefHeading___Toc566_4081095476)

[4.6 Mehrere Tabellen Ohne Inner Join III 8](#__RefHeading___Toc568_4081095476)

[4.7 Mehrere Tabellen Ohne Inner Join IV 8](#__RefHeading___Toc570_4081095476)

[4.8 Inner Join I 9](#__RefHeading___Toc572_4081095476)

[4.9 Inner Join II 9](#__RefHeading___Toc574_4081095476)

[4.10 Inner Join III 10](#__RefHeading___Toc576_4081095476)

[4.11 Inner Join IV 10](#__RefHeading___Toc578_4081095476)

[4.12 Aggregatsfunktion Min 11](#__RefHeading___Toc580_4081095476)

[4.13 Aggregatsfunktion Max 11](#__RefHeading___Toc582_4081095476)

[4.14 Aggregatsfunktion Count 12](#__RefHeading___Toc584_4081095476)

[4.15 Aggregationsfunktion Sum 12](#__RefHeading___Toc586_4081095476)

[4.16 Aggregatsfunktion Avarage 13](#__RefHeading___Toc588_4081095476)

[4.17 Tabelle Erstellen 13](#__RefHeading___Toc590_4081095476)

[4.18 Spieldauer speichern 16](#__RefHeading___Toc592_4081095476)

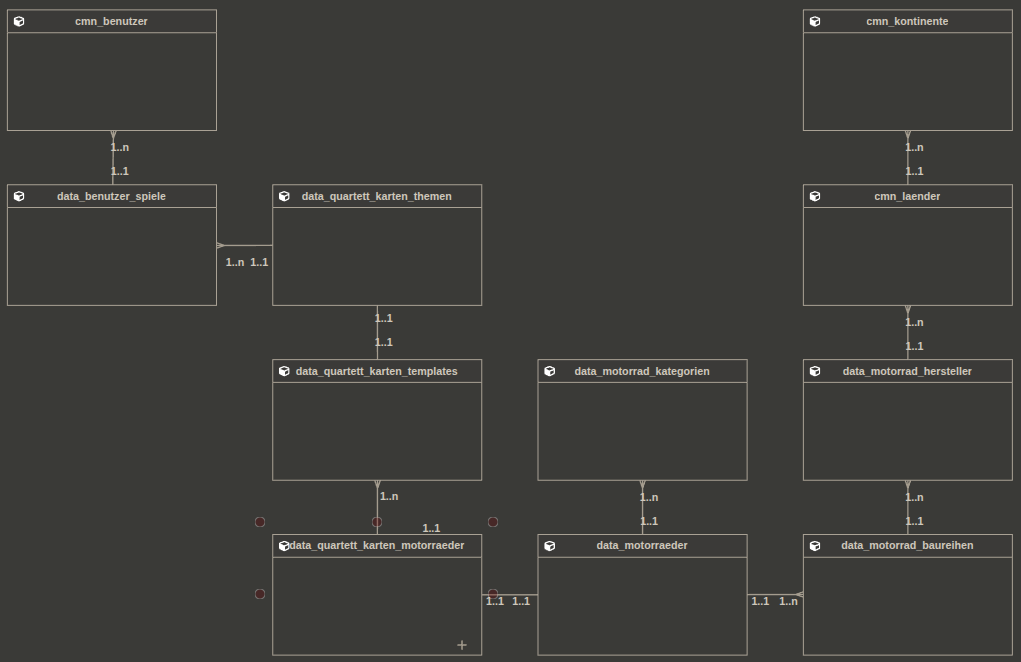
[4.19 Löschen 17](#__RefHeading___Toc594_4081095476)

[4.20 Hinzufügen eines Attributes 17](#__RefHeading___Toc596_4081095476)

# Datenbank Beschreibung

Die Datenbank hat das Ziel Daten für ein Quartett Kartenspiel zu speichern. Das Spiel ermöglicht es Nutzern Quartett zu verschiedenen Themen gegeneinander zu Spielen und sein Wissen darin zu testen.

# ER-Modell

Abbildung 1: Vereinfachtes ER-Modell

# Relationsschreibweise

Hinweis:

Jede Tabelle hat zusätzlich folgende Audit-Spalten (ersteller, ersteller\_am, aktualisiert\_von, aktualisert\_am). Der einfachheit halber werden diese nicht weiter aufgeführt.

cmn\_benutzer (benutzer\_id, name, email\_adresse, passwort)

data\_benutzer\_spiele (benutzer\_spiele\_id, *quartett\_karten\_thema\_id*, *spieler\_1\_id,spieler\_2\_id, spieler\_3\_id, spieler\_4\_id, gewinner\_id)*

data\_quartett\_karten\_themen (quartett\_karten\_thema\_id, name, beschreibung)

data\_quartett\_karten\_templates (quartett\_karten\_template\_id, *quartett\_karten\_thema\_id*, karten\_attribute)

data\_quartett\_karten\_motorraeder (quartett\_karten\_motorrad\_id, *quartett\_karten\_template\_id, motorrad\_id*)

data\_motorraeder (motorrad\_id, *motorrad\_hersteller\_id, motorrad\_baureihe\_id,motorrad\_kategorie\_id,* name, baujahr, leistung\_kw, drehmomet\_nm, 0\_auf\_100\_s, hoechstgeschwindigkeit\_kmh, verbrauch\_kml, laenge\_mm, breite\_mm, hoehe\_mm, sitzhoehe\_mm, gewicht\_kg, preis\_euro, bild, bild\_mime\_type)

data\_motorrad\_kategorien (motorrad\_kategorie\_id, name, beschreibung)

data\_motorrad\_baureihen (motorrad\_baureihe\_id, *motorrad\_hersteller\_id*, bezeichnung, beschreibung, motortyp, hubraum\_ccm, baujahr\_beginn, baujahr\_ende, vorgaenger\_id)

data\_motorrad\_hersteller (motorrad\_hersteller\_id, name, beschreibung, *herkunfsland\_id,* gruendungsjahr, aktiv, logo, logo\_mime\_type)

cmn\_laender (land\_id, name, iso\_code\_2, *kontinent\_id,* bevoelkerung, land\_flaeche\_qkm, breitengrad, laengengrad)

cmn\_kontinente (kontinent\_id, name)

# SQL-Statement

## Eine Tabelle I

Beschreibung:

Auflistung der E-Mail Adressen aller Nutzer.

SQL:

select email\_adresse

from cmn\_benutzer

order by email\_adresse;

## Eine Tabelle II

Beschreibung:

Auflistung aller Motorradkategorien.

SQL:

select name

from data\_motorrad\_kategorien

order by name;

## Eine Tabelle III

Beschreibung:

Auflistung aller Motorräder mit "GSX S1000" im Namen.

SQL:

select name

from data\_motorraeder

where name like '%GSX S1000%'

order by name;

## Mehrere Tabellen Ohne Inner Join I

Beschreibung:

Auflistung der Gewinner aller Spiele.

SQL:

select

bs.benutzer\_spiel\_id as spiel\_nummer,

b.name as gewinner

from cmn\_benutzer b, data\_benutzer\_spiele bs

where b.benutzer\_id = bs.gewinner\_id

order by bs.benutzer\_spiel\_id;

## Mehrere Tabellen Ohne Inner Join II

Beschreibung:

Auflistung der Kartenatribute aller Themen.

SQL:

select

qkth.name as thema,

qkte.karten\_attribute as attribute

from data\_quartett\_karten\_themen qkth, data\_quartett\_karten\_templates qkte

where qkth.quartett\_karten\_thema\_id = qkte.quartett\_karten\_thema\_id

order by qkth.name;

## Mehrere Tabellen Ohne Inner Join III

Beschreibung:

Auflistung aller Motorrad Baureihen, welche noch produziert werden.

SQL:

select

mh.name as hersteller,

mb.bezeichnung as baureihe,

mb.baujahr\_beginn,

mb.baujahr\_ende

from data\_motorrad\_hersteller mh, data\_motorrad\_baureihen mb

where mh.motorrad\_hersteller\_id = mb.motorrad\_hersteller\_id

and mb.baujahr\_ende is null;

## Mehrere Tabellen Ohne Inner Join IV

Beschreibung:

Auflistung aller Motorräder geordnet nach Kategorie.

SQL:

select

mk.name as kategorie,

m.name as motorrad

from data\_motorraeder m, data\_motorrad\_kategorien mk

where m.motorrad\_kategorie\_id = mk.motorrad\_kategorie\_id

order by mk.name, m.name;

## Inner Join I

Beschreibung:

Auflistung aller Kontinente und Länder.

SQL:

select

k.name as kontinent,

l.name as land

from cmn\_laender l

inner join cmn\_kontinente k on l.kontinent\_id = k.kontinent\_id;

## Inner Join II

Beschreibung:

Auflistung der Spielteilnehmer aller Spiele.

SQL:

select

bs.benutzer\_spiel\_id as spiel\_nummer,

b1.name as spieler\_1,

b2.name as spieler\_2,

b3.name as spieler\_3,

b4.name as spieler\_4

from data\_benutzer\_spiele bs

inner join cmn\_benutzer b1 on bs.spieler\_1\_id = b1.benutzer\_id

inner join cmn\_benutzer b2 on bs.spieler\_2\_id = b2.benutzer\_id

inner join cmn\_benutzer b3 on bs.spieler\_3\_id = b3.benutzer\_id

inner join cmn\_benutzer b4 on bs.spieler\_4\_id = b4.benutzer\_id

order by bs.benutzer\_spiel\_id;

## Inner Join III

Beschreibung:

Auflistung der Themen aller Spiele.

SQL:

select

bs.benutzer\_spiel\_id as spiel\_nummer,

qkt.name

from data\_benutzer\_spiele bs

inner join data\_quartett\_karten\_themen qkt on bs.quartett\_karten\_thema\_id = qkt.quartett\_karten\_thema\_id

order by bs.benutzer\_spiel\_id;

## Inner Join IV

Beschreibung:

Alle Motorräder aus Asien sollen aufgelistet werden.

SQL:

select

m.name as motorrad

from data\_motorraeder m

inner join data\_motorrad\_baureihen mb on m.motorrad\_baureihe\_id = mb.motorrad\_baureihe\_id

inner join data\_motorrad\_hersteller mh on m.motorrad\_hersteller\_id = mh.motorrad\_hersteller\_id

inner join cmn\_laender l on mh.herkunftsland\_id = l.land\_id

inner join cmn\_kontinente k on l.kontinent\_id = k.kontinent\_id

where k.name = 'Asien';

## Aggregatsfunktion Min

Beschreibung:

Das Motorrad mit der geringsten Leistung soll ermittelt werden.

SQL:

select

name,

min(leistung\_kw)

from data\_motorraeder

group by name, leistung\_kw

order by leistung\_kw asc

limit 1;

## Aggregatsfunktion Max

Beschreibung:

Das teuerste Motorrad soll ermittelt werden.

SQL:

select

mh.name as hersteller,

max(m.preis\_euro)

from data\_motorraeder m

join data\_motorrad\_hersteller mh on m.motorrad\_hersteller\_id = mh.motorrad\_hersteller\_id

group by mh.name, m.preis\_euro

order by m.preis\_euro desc

limit 1;

## Aggregatsfunktion Count

Beschreibung:

Zählen der Siege aller Spiele.

SQL:

select

b.name as gewinner,

count(1) as siege

from data\_benutzer\_spiele bs

join cmn\_benutzer b on bs.gewinner\_id = b.benutzer\_id

group by b.name

order by b.name;

## Aggregationsfunktion Sum

Beschreibung:

Die Gesamtbevölkerung in Südamerika soll berechnet werden.

SQL:

select

sum(bevoelkerung) gesamt\_bevoelkerung

from cmn\_laender l

inner join cmn\_kontinente k on l.kontinent\_id = k.kontinent\_id

where k.name = 'Südamerika';

## Aggregatsfunktion Avarage

Beschreibung:

Die durschnitssgeschwindigkeit aller Supersportler soll ermittelt werden.

SQL:

select

avg(hoechstgeschwindigkeit\_kmh)

from data\_motorraeder m

join data\_motorrad\_kategorien mk on m.motorrad\_kategorie\_id = mk.motorrad\_kategorie\_id

where mk.name = 'Supersportler';

## Tabelle Erstellen

Beschreibung:

Es wird eine Tabelle zum Speichern von Motorrädern erstellt.

SQL:

create table data\_motorraeder (

motorrad\_id int auto\_increment not null comment 'Primärschlüssel des Motorrads',

ersteller varchar(256) not null default 'DATABASE' comment 'Nutzer welcher diesen Eintrag erstellt hat',

erstellt\_am timestamp not null default current\_timestamp comment 'Zeitstempel des Erstellungsdatums',

aktualisiert\_von varchar(256) default 'DATABASE' comment 'Nutzer welcher diesen EIntrag zuletzt aktualisiert hat',

aktualisiert\_am timestamp default current\_timestamp on update current\_timestamp comment 'Zeitstempel des letzten Aktualisierungsdatums',

motorrad\_hersteller\_id int not null comment 'Id des Herstellers des Motorrads. Fremdschlüssel von data\_hersteller',

motorrad\_baureihe\_id int not null comment 'Id der Baureihe des Motorrads. Fremdschlüssel von data\_baureihen',

motorrad\_kategorie\_id int not null comment 'Id der Kategorie des Motorrads. Fremdschlüssel von data\_kategorien',

name varchar(256) not null comment 'Name des Motorrads',

baujahr int not null comment 'Baujahr des Motorrads',

leistung\_kw decimal(5,2) comment 'Leistung (kw) des Motorrads',

drehmoment\_nm decimal(6,2) comment 'Drehmoment (nm) des Motorrads',

0\_auf\_100\_s decimal(5,2) comment 'Beschleuniging von 0 auf 100km/h (s) des Motorrads',

hoechstgeschwindigkeit\_kmh int comment 'Höchstgeschwindigkeit (km/h) des Motorrads',

verbrauch\_kml decimal(5,2) comment 'Verbrauch (kilometer/liter) des Motorrads',

laenge\_mm int comment 'Größe (mm) des Motorrads',

breite\_mm int comment 'Breite (mm) des Motorrads',

hoehe\_mm int comment 'Höhe (mm) des Motorrads',

sitzhoehe\_mm int comment 'Sitzhöhe (mm) des Motorrads',

gewicht\_kg decimal(6,2) comment 'Gewicht (kg) des Motorrads',

preis\_euro decimal(10,2) comment 'Preis (€) des Motorrads',

bild blob comment 'Logo des Motorrads',

bild\_mime varchar(16) comment 'Mime type des Bildes des Motorrads. Erlaubt sind JPG und PNG',

constraint data\_motorraeder\_pk primary key (motorrad\_id),

constraint data\_motorraeder\_motorrad\_hersteller\_fk

foreign key (motorrad\_hersteller\_id) references data\_motorrad\_hersteller (motorrad\_hersteller\_id)

on update cascade

on delete cascade,

constraint data\_motorraeder\_motorrad\_baureihen\_fk

foreign key (motorrad\_baureihe\_id) references data\_motorrad\_baureihen (motorrad\_baureihe\_id)

on update cascade

on delete cascade,

constraint data\_motorraeder\_motorrad\_kateogrien\_fk

foreign key (motorrad\_kategorie\_id) references data\_motorrad\_kategorien (motorrad\_kategorie\_id)

on update cascade

on delete cascade

) comment = 'Tabelle zum speichern von Daten über Motorräder.';

create index idx\_data\_motorraeder\_motorrad\_hersteller\_id on data\_motorraeder (motorrad\_hersteller\_id);

create index idx\_data\_motorraeder\_motorrad\_baureihe\_id on data\_motorraeder (motorrad\_baureihe\_id);

create index idx\_data\_motorraeder\_motorrad\_kategorie\_id on data\_motorraeder (motorrad\_kategorie\_id);

create index idx\_data\_motorraeder\_name on data\_motorraeder (name);

create index idx\_data\_motorraeder\_baujahr on data\_motorraeder (baujahr);

create index idx\_data\_motorraeder\_leistung\_kw on data\_motorraeder (leistung\_kw);

create index idx\_data\_motorraeder\_drehmoment\_nm on data\_motorraeder (drehmoment\_nm);

create index idx\_data\_motorraeder\_verbrauch\_kml on data\_motorraeder (verbrauch\_kml);

create index idx\_data\_motorraeder\_0\_auf\_100\_s on data\_motorraeder (0\_auf\_100\_s);

create index idx\_data\_motorraeder\_hoechstgeschwindigkeit\_kmh on data\_motorraeder (hoechstgeschwindigkeit\_kmh);

create index idx\_data\_motorraeder\_sitzhoehe\_mm on data\_motorraeder (sitzhoehe\_mm);

create index idx\_data\_motorraeder\_gewicht\_kg on data\_motorraeder (gewicht\_kg);

create index idx\_data\_motorraeder\_preis\_euro on data\_motorraeder (preis\_euro);

## Spieldauer speichern

Beschreibung:

Für die Statistik soll auch noch die Spieldauer gespeichert werden.

SQL:

alter table data\_benutzer\_spiele

add column spiel\_dauer\_min int not null;

## Löschen

Beschreibung:

Der Nutzer mit der Id 7 soll gelöscht werden.

SQL:

delete from cmn\_benutzer where benutzer\_id = 7;

## Hinzufügen eines Attributes

Beschreibung:

Ein weiterer Hersteller wird in die Motorrad Hersteller Tabelle hinzugefügt.

SQL:

insert into data\_motorrad\_hersteller (

motorrad\_hersteller\_id,

name,

beschreibung,

herkunftsland\_id,

gruendungsjahr,

aktiv

) values (

15,

'Royal Enfield',

'Royal Enfield ist ein indischer Motorradhersteller und zugleich die älteste noch produzierte Motorradmarke der Welt. Das 1955 in seiner heutigen Form gebildete Unternehmen ist aus dem gleichnamigen britischen Motorradhersteller hervorgegangen, der auch Automobile und Gewehre produzierte. Seit 1994 gehört das Unternehmen als Tochtergesellschaft zum indischen Konzern Eicher.',

76,

1901,

'J'

);