

# INSTAHUB / SQL: AGGREGATSFUNKTIONEN

Unter *Aggregatsfunktionen* versteht man Funktionen, die Daten in irgendeiner Form zusammenfassen. Hier die wichtigsten Arten Aggregatsfunktionen:

- Anzahl
- Mittelwert
- Minimum / Maximum
- Summe

## Anzahl

**SELECT COUNT(<Attribut>) AS <Name der Anzahl-Spalte>  
FROM <Tabelle>;**

*Bsp.: Es soll die Anzahl der Benutzer (=users) ermittelt werden.*

**SELECT COUNT(id) AS Anzahl\_der\_Benutzer  
FROM users;**



Wir zählen hier ab, wie viele *Benutzer-IDs* es gibt. Alternativ könnte man auch abzählen, wie viele *Zeilen* es in der Tabelle *users* gibt. Das geht so:

**SELECT COUNT(\*) AS Anzahl\_der\_Benutzer  
FROM users;**

users	
id	INT
username	VARCHAR(45)
email	VARCHAR(45)
password	VARCHAR(45)
name	VARCHAR(45)
bio	VARCHAR(45)
gender	VARCHAR(45)
birthday	DATE
city	VARCHAR(45)

Anfrage ausgeführt. 1 Ergebnisse gefunden.	
Anzahl_der_Benutzer	204

## Mittelwert (=Average)

**SELECT AVG(<Attribut>) AS <Name der Mittelwert-Spalte>  
FROM <Tabelle>;**

*Bsp.: Es soll die durchschnittliche Körpergröße aller Benutzer ermittelt werden.*

**SELECT AVG(centimeters) AS Durchschnittsgröße  
FROM users;**

Durchschnittsgröße
170.3202



Es gibt mit Sicherheit sinnvollere Abfragen, als die nach der durchschnittlichen Körpergröße der User. Für die „sinnvolleren“ Abfragen brauchen wir bei Instahub aber die Daten aus mehreren Tabellen, was wir noch nicht beherrschen. Wir werden in späteren Aufgaben sinnvollere Anwendungsmöglichkeiten für den Mittelwert und die anderen Aggregatsfunktionen finden!

## Minimum / Maximum

**SELECT MIN(<Attribut>) AS <Name der Minimum-Spalte>  
FROM <Tabelle>;**

*Bsp.: Finden Sie heraus, wann der älteste Benutzer geboren wurde!  
(Der älteste Benutzer hat das früheste, also „kleinste“ Geburtsdatum.)*

**SELECT MIN(birthday) AS Ältester\_Benutzer  
FROM users;**

Ältester_Benutzer
1938-03-21 00:00:00



Das Maximum funktioniert genauso. Hier wird MIN durch MAX ersetzt!

## Summe



Für die Summe gibt es in Instahub leider keine einfachen Beispiele. Damit man sinnvoll eine Summe bilden kann, ist es nämlich erforderlich, dass in einer Tabelle *Zahlen* stehen, bei denen das „zusammenzählen“ Sinn macht. Was bei Instahub leider nicht der Fall ist. Wir schauen uns die Summen-Funktion deshalb mit einem leider völlig sinnfreien Beispiel an, damit wir für Abfragen mit anderen Datenbanken gerüstet sind!

**SELECT SUM(<Attribut>) AS <Name der Summen-Spalte>  
FROM <Tabelle>;**

*Bsp.: Zählen Sie die Körpergrößen aller Benutzer zusammen!*

**SELECT SUM(centimeters) AS Sinnlose\_Summe  
FROM users;**

## Aufgaben:



	<i>Erstellen Sie ein Word-Dokument, in dem Sie Ihre Lösungen abspeichern und später ausdrucken können.</i> <b>(Für die Abfragen benötigen Sie eine dieser Tabellen: photos, users)</b>	
<b>1.</b>	Ermitteln Sie mit einer Abfrage, wie viele Fotos in Instahub gespeichert sind! <i>(Lösung: 1.522)</i>	
<b>2.</b>	Wie viele Bilder hat der Benutzer mit der Nummer 10 (user_id) gespeichert? <i>(Lösung: 8)</i>	
<b>3.</b>	Wie viele weibliche Mitglieder gibt es? <i>(Lösung: 104)</i>	
<b>4.</b>	Welche Körpergröße hat der größte Benutzer? <i>(Lösung: 190)</i>	

## Gruppierungen mit GROUP BY

Mit der GROUP BY – Klausel werden gleiche Daten zusammengefasst.

```
SELECT <Attribut>
FROM <Tabelle>
GROUP BY <Attribut>;
```

Bsp.:

Es sollen alle Städte angezeigt werden, die die Benutzer als Wohnort angegeben haben. Hinter jeder Stadt soll angezeigt werden, wie viele Benutzer in dieser Stadt wohnen. Dabei soll jede Stadt nur einmal aufgelistet werden.

```
SELECT city, COUNT(*) AS Anzahl
FROM users
GROUP BY city;
```

Albertshofen	1
Angersdorf	1
Bad Essen	2

## Aufgaben:



<b>5.</b>	Lassen Sie sich in einer Abfrage anzeigen, wie viele unserer Benutzer männlich bzw. weiblich sind. <small>(Hinweis: Weil Sie selbst auch ein Nutzer sind, kann es sein, dass Ihre Zahlen um 1 abweichen!)</small>	<table border="1"><thead><tr><th>Geschlecht</th><th>Anzahl</th></tr></thead><tbody><tr><td>male</td><td>100</td></tr><tr><td>female</td><td>104</td></tr></tbody></table>	Geschlecht	Anzahl	male	100	female	104										
Geschlecht	Anzahl																	
male	100																	
female	104																	
<b>6.</b>	Für jeden Benutzer soll angezeigt werden, wie viele Bilder er besitzt. Sortieren Sie diese Liste nach der Anzahl der Bilder absteigend.	<table border="1"><thead><tr><th>user_id</th><th>Anzahl_Bilder</th></tr></thead><tbody><tr><td>111</td><td>16</td></tr><tr><td>43</td><td>16</td></tr><tr><td>105</td><td>15</td></tr></tbody></table>	user_id	Anzahl_Bilder	111	16	43	16	105	15								
user_id	Anzahl_Bilder																	
111	16																	
43	16																	
105	15																	
<b>7.</b>	Finden Sie heraus, wie viele Kommentare der Benutzer mit der Nummer 10 ( <i>user_id</i> ) bisher verfasst hat. <small>(Hierfür benötigen Sie die Tabelle <i>comments</i>)</small>	<table border="1"><thead><tr><th>user_id</th><th>Anzahl_Kommentare</th></tr></thead><tbody><tr><td>10</td><td>9</td></tr></tbody></table> <table border="1"><thead><tr><th colspan="2">comments</th></tr></thead><tbody><tr><td>id</td><td>INT</td></tr><tr><td>user_id</td><td>INT</td></tr><tr><td>photo_id</td><td>INT</td></tr><tr><td>body</td><td>VARCHAR(255)</td></tr><tr><td>commentscol</td><td>VARCHAR(45)</td></tr></tbody></table>	user_id	Anzahl_Kommentare	10	9	comments		id	INT	user_id	INT	photo_id	INT	body	VARCHAR(255)	commentscol	VARCHAR(45)
user_id	Anzahl_Kommentare																	
10	9																	
comments																		
id	INT																	
user_id	INT																	
photo_id	INT																	
body	VARCHAR(255)																	
commentscol	VARCHAR(45)																	