1 Einfache SQL-Abfragen

1.1 SQL kennen lernen

SQL (*structured query language*) ist eine Programmiersprache, mit der man auf Datenbanken zugreifen kann. SQL ist eine Vertreterin der so genannten **deklarativen Programmierung**, da man mit ihr nur das Ergebnis beschreibt, nicht aber den Weg dorthin.

1.1.1 Die erste SQL-Abfrage

1. *Rufen* Sie Ihren InstaHub auf und melden Sie sich als `Admin` dort an!
2. Klicken oben rechts auf das Symbol *Datenbank* und wählen Sie `SQL`:

Abbildung 1: Den SQL-Editor öffnen

3. Sie gelangen in den SQL-Editor:

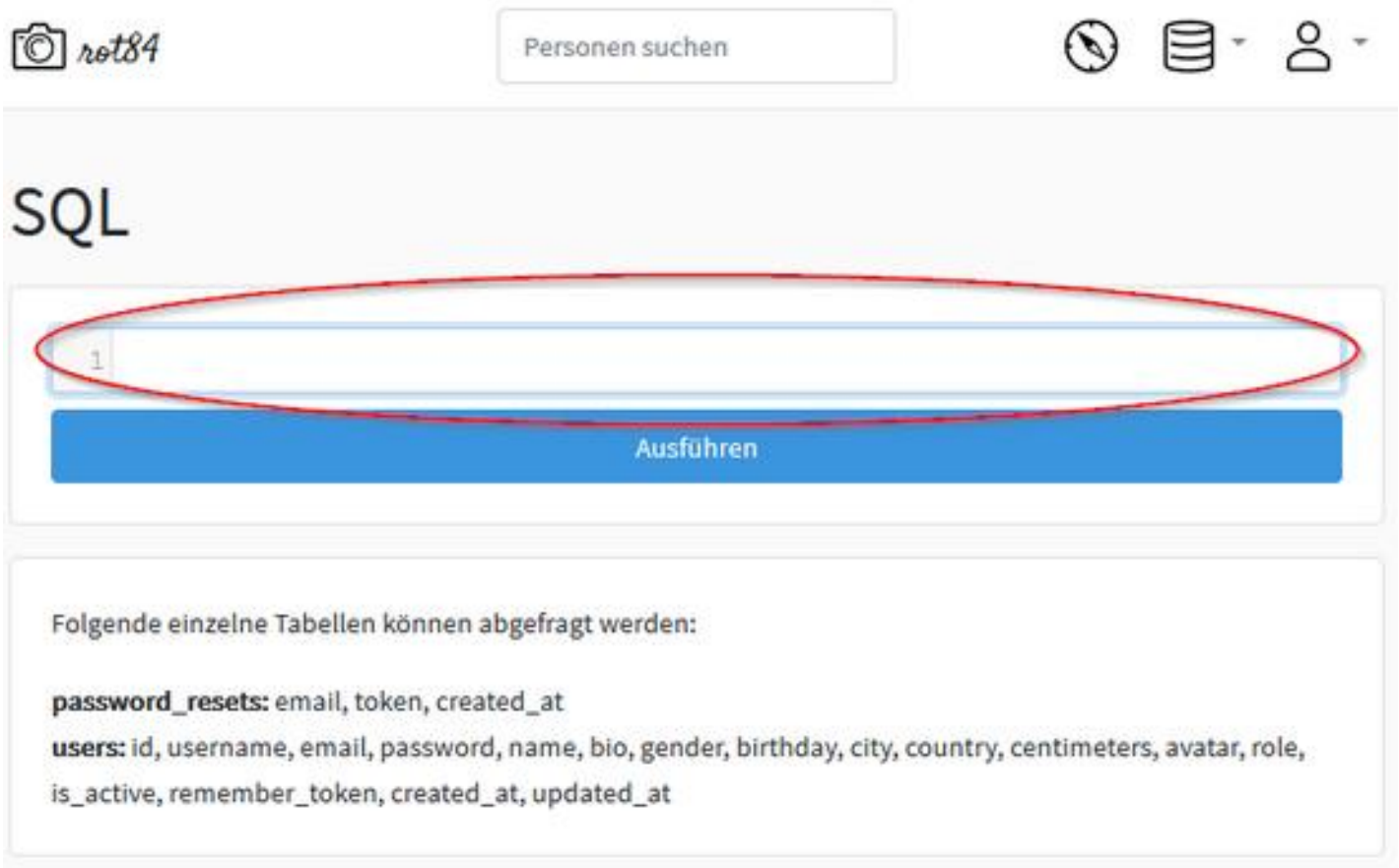

Abbildung 2: Der SQL-Editor

4. Setzen Sie den Cursor in das leere, im obigen Screenshot rot markierten, Eingabefeld und tippen Sie die folgende SQL-Abfrage exakt so ein:

```
SELECT   username, name, birthday, city
FROM users
```

   Tipps: Nach dem `SELECT` bzw. `FROM` tippen Sie die Tab-Taste, damit der Befehl übersichtlich aussieht. Am Ende der ersten Zeile können Sie `ENTER` drücken, um in eine neue Zeile zu wechseln.
5. Klicken Sie auf `Ausführen`! Sie sehen nun eine Ergebnistabelle mit gut 200 Zeilen:

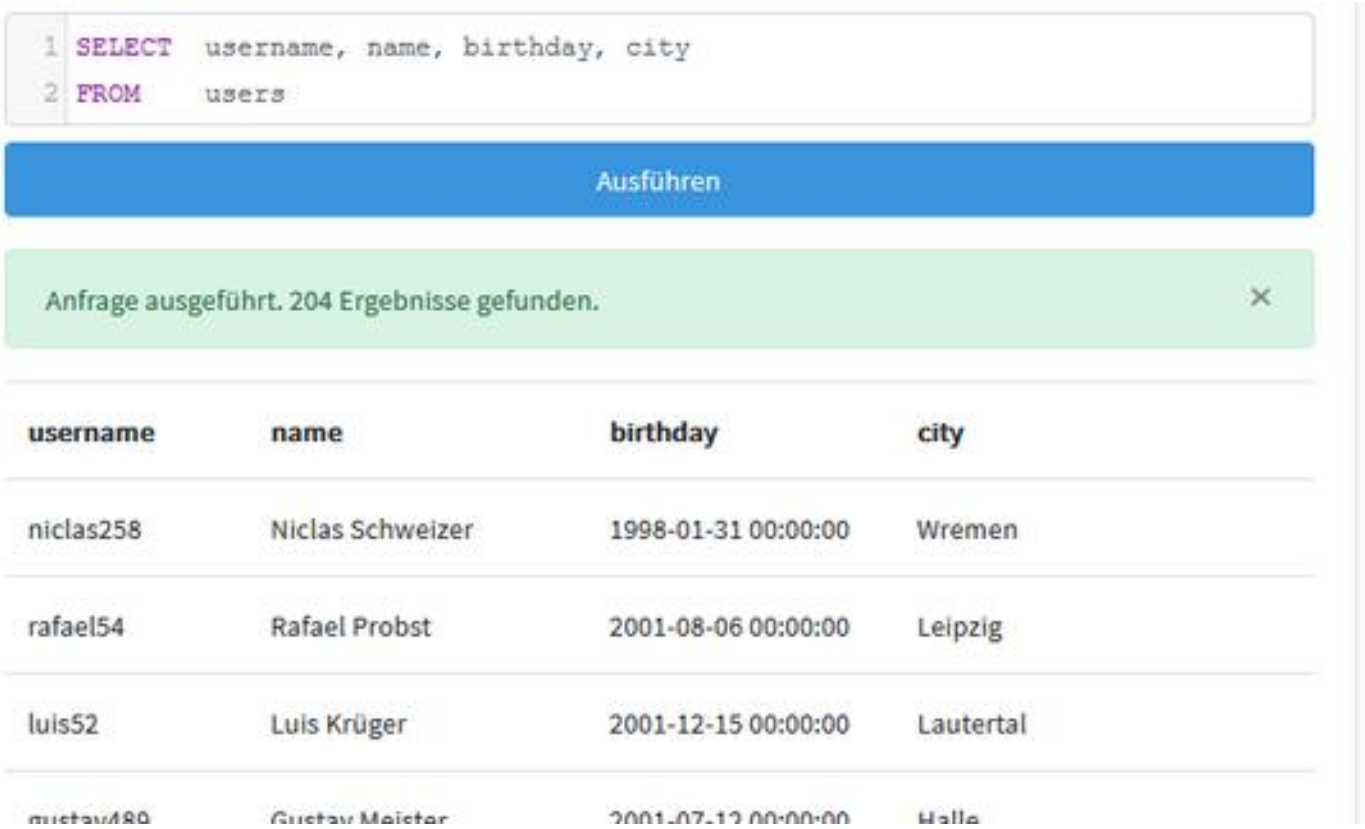

Abbildung 3: Das Ergebnis Ihres ersten SELECT-Befehls

1.1.2 Tipps:

- Beachten Sie, dass die Spaltennamen exakt mit den Bezeichnungen nach dem SELECTübereinstimmen!
- SQL unterscheidet nicht zwischen Groß- und Kleinschreibung. Es ist aber üblich, die SQL-Befehle in GROSSBUCHSTABEN zu notieren. Halten Sie sich bitte an diese Gepflogenheit.
- Wenn Sie *alle* Spalten ausgeben wollen, können Sie statt alle Spalten aufzuzählen auch einfach ein * notieren, z. B.;

```
SELECT    *
FROM      users
```

- Wenn Sie statt eine Antwort wie der obigen eine englische Fehlermeldung bekommen, haben Sie bei der Eingabe des Befehls einen Fehler gemacht. Kontrollieren Sie noch einmal ganz genau, ob Sie irgendwo einen kleinen Fehler gemacht haben. Sind die Kommata wirklich da, wo sie sein sollen und sind es wirklich Kommata? Kein kleiner Tippfehler in den Spaltennamen?

1.2 SQL-Abfragen verschönern und sortieren

Wenn Sie die Ergebnistabelle ausdrucken wollen, kann es hilfreich sein, die

Spaltenbezeichnungen verändern zu können. *Probieren* Sie aus, was passiert, wenn Sie statt `birthday` den Text `birthday AS "Geburtstag"`eintragen!

Außerdem können Sie die Klauseln `ORDER BY` und `LIMIT` verwenden:

```
-- Verschoenerte Abfrage
SELECT  username, name, birthday AS "Geburtstag",
        city AS "Stadt"
FROM    users
ORDER BY birthday ASC
LIMIT   10
```

1.2.1 Kommentare

Die erste Zeile beginnt mit zwei Minuszeichen (`--`). Diese bedeuten, dass alle Zeichen, die bis zum Ende der Zeile kommen, überlesen werden. Sie können Kommentare z. B. verwenden, um deutlich zu machen, welche Aufgabe Sie bearbeiten.

Wenn Sie mehrere Zeilen als Kommentar markieren wollen, verwenden Sie `/*` und `*/`:

```
/* Verschoenerte Abfrage
   Liefert alle Nutzer*innen nach Geburtstag sortiert. */
SELECT  username, name, birthday AS "Geburtstag",
        city AS "Stadt"
FROM    users
ORDER BY birthday ASC
LIMIT   10
```

1.2.2 ORDER und LIMIT

Statt ASC können Sie in der ORDER-Klausel auch `DESC` oder gar nichts verwenden. *Beschreiben* Sie, was jeweils passiert!

Beschreiben Sie, welche Bedeutung die `LIMIT`-Klausel hat!

Aufgabe 5.1: Erste SQL-Abfragen

Erstellen Sie jeweils eine SQL-Abfrage, die die folgenden verbalen Anfragen möglichst schön beantworten:

1. Wer sind die zehn jüngsten InstaHub-User*Innen?
2. Wo wohnen die fünf am längsten angemeldeten InstaHub-User*Innen?
3. Wer ist der erste InstaHub-User*in gewesen?

1.3 Dubletten aussortieren

Uns interessiert brennend, in welchen Städten wir bereits mindestens eine/n Nutzer*In haben. Wir probieren es mit der folgenden Abfrage:

```
SELECT  city
FROM    users
ORDER BY city ASC
```

Das Ergebnis ist eine lange Liste von Städten, in denen aber einige Städte mehrfach vorkommen.

Probieren Sie es einmal mit

```
SELECT  DISTINCT city
FROM    users
ORDER BY city ASC
```

Nun werden alle Zeilen aussortiert, die mehr als einmal vorkommen.

Aufgabe 5.2: Dubletten aussortieren

Erstellen Sie jeweils eine SQL-Abfrage, die folgenden verbalen Anfragen möglichst schön beantworten:

1. Welche Körpergrößen haben unser Nutzer*innen?
2. Welche Rollen haben unsere Nutzer*Innen?

1.4 Datensätze filtern

Welche InstaHub-Nutzer*Innen leben eigentlich in Dresden?

Wir könnten nun die Liste nach Städten sortieren und dann scrollen, bis wie bei *Dresden* angekommen sind. Das muss einfach gehen. Na klar. Die wohl wichtigste Klausel des `SELECT`-Befehls fehlt uns noch: `WHERE`

```
SELECT  username, name, city
WHERE   city = "Dresden"
```

Aufgabe 5.3: Deutschlandreise I

Erstellen Sie jeweils eine SQL-Abfrage, die folgenden verbalen Anfragen möglichst schön beantworten:

1. Welche Nutzer*Innen wohnen in Leipzig?
2. Welche Nutzer*Innen haben die Rolle `dba`?
3. Welche Nutzer*Innen wohnen in Bokholt-Hanredder?
4. Welche Nutzer*Innen wohnen in Hamburg?
5. Ist Justin Schuster bei uns angemeldet?

Aufgabe 5.4 Ups…
Überprüfen Sie Ihre Ergebnisse aus den beiden letzten Abfragen der vorangegangenen Aufgabe noch einmal genau und *begründen* Sie, warum Ihre Ergebnisse offenbar nicht ganz korrekt sind!

1.4.1 Zeichenketten vergleichen

Beim Vergleichen von Zeichenketten ist es oft hilfreich, wenn man nach Teilzeichenketten suchen kann. Hierfür hat SQL den Operator `LIKE`, der anders als = nicht die exakte Übereinstimmung erfordert, sondern mit Hilfe von Wildcards auch Wortteile finden kann. So findet `WHERE name LIKE "Justin%Schuster%"` auch die folgenden Personen:

- Justine Schuster
- Justin Frederik Augustus Frederikus Freiherr von Boomzwickel-Schuster

Während mit % beliebig viele Zeichen (inklusive keinem) gefunden werden, wird mit dem Zeichen _ (Unterstrich, Shift+Minuszeichen) nur genau ein Zeichen gefunden.

Aufgabe 5.5 Deutschlandreise II
Erstellen Sie jeweils eine SQL-Abfrage, die folgenden verbalen Anfragen möglichst schön beantworten:

1. Welche Nutzer*Innen wohnen in Hamburg?
2. Welche Nutzer*Innen mit dem Namen Schuster sind bei uns angemeldet?
3. Welche Nutzer*Innen mit dem haben wir, die so klingen wie Meier? (Hier können Sie einen Beifang irrelevanter Datensätze wohl nicht völlig verhindern!)

1.4.2 Numerische Vergleiche

Mit den Vergleichsoperatoren >, >=, < und <=können auch numerische Werte verglichen werden.

Das funktioniert auch mit Kalenderdaten. `WHERE created_at < "2017-12-31"` liefert alle Benutzer*Innen, deren Datensätze vor dem 31.12.2017 erstellt wurden.

Aufgabe 5.6 Riesen und Zwerge
Erstellen Sie jeweils eine SQL-Abfrage, die folgenden verbalen Anfragen möglichst schön beantworten:

1. Welche InstaHub-Nutzer*Innen sind größer als 190cm?

2. Welche InstaHub-Nutzer*Innen sind kleiner als 150cm?
3. Welche InstaHub-Nutzer*Innen sind nicht volljährig?
4. Welche InstaHub-Nutzer*Innen sind älter als 70 Jahre?
5. Gibt es eine/n Nutzer*In, die am selben Tag Geburtstag hat wie Sie?