

11.10.2022

## Übungsblatt 1 – Lösungsvorschlag

### Diskussionsgrundlage Proseminar

In diesem Semester werden wir die Proseminar-Stunden intensiv für das gemeinsame Diskutieren und Lösen von Beispielen verwenden. Daher enthält jedes Übungsblatt zunächst einen Abschnitt mit Beispielen und Aufgaben. Dieser sollte als Diskussionsgrundlage für das Proseminar dienen. Die Beispiele werden wir während des Proseminars lösen, sie dienen als Vorbereitung für die zu lösenden Aufgaben und Klausuren.

Die Proseminar-Stunden sollen für Sie ein Forum darstellen. Lösen Sie die im Proseminar gestellten Aufgaben in Gruppen und diskutieren Sie die verschiedenen Möglichkeiten. Gestalten Sie mit, fragen Sie Ihre Proseminar-Leiterin/Ihren Proseminar-Leiter, Ihre Kolleginnen und Kollegen und nützen Sie die Zeit, um sich intensiv über die Inhalte auszutauschen.

*Wichtig:* Stellen Sie Fragen (auch wenn sie ihnen noch so “blöd” erscheinen), nur so können wir die noch offenen Themen und Fragen gemeinsam beantworten. Für diese Aufgaben und Diskussionen werden keinerlei Noten vergeben.

### Tools

Im Folgenden beschreiben wir kurz jene Tools, die Sie für die Bearbeitung der Aufgaben benötigen werden.

### Diagramme

In diesem Proseminar werden wir häufig ER-Diagramme zeichnen. Als Editor für Diagramme können wir zum Beispiel yEd<sup>1</sup> empfehlen. Sollten Sie sich für diesen Editor entscheiden, so können Sie auf das Template `ER-master.graphmlOLAT` zurückgreifen, um schneller einzelne ER-Elemente zu zeichnen. Selbstverständlich können Sie auch gerne andere Editoren wie Draw.io<sup>2</sup> verwenden oder Zeichnungen von Hand anfertigen (diese können Sie für die Abgabe von Lösungen fotografieren).

### Datenbankmanagementsystem

Als Datenbankmanagementsystem (DBMS) verwenden wir PostgreSQL 13<sup>3</sup>. Alle Aufgaben wurden auf diesem System erstellt und getestet. Von einer Verwendung von MySQL/MariaDB<sup>4</sup> raten wir für dieses Proseminar ab, da sich die Funktionalität und die SQL-Syntax von PostgreSQL und MySQL/-MariaDB leicht unterscheiden. Es steht Ihnen natürlich frei, für die Ausarbeitung Ihrer Lösungen ein beliebiges DBMS zu verwenden, solange Ihre Abgaben auf PostgreSQL 13 ausführbar sind.

---

<sup>1</sup><https://yworks.com/products/yed>

<sup>2</sup><https://about.draw.io>

<sup>3</sup><https://postgresql.org/docs/13/index.html>

<sup>4</sup><https://mariadb.org>

## SQL Client

Als SQL Client können wir Ihnen pgAdmin<sup>5</sup> empfehlen. Natürlich steht es Ihnen auch hier frei, ein anderes Tool wie zum Beispiel DBeaver<sup>6</sup> zu wählen.

## Docker (Optional)

Sie können gerne das Datenbankmanagementsystem und einen SQL Client ihrer Wahl direkt auf ihrem System installieren. Um das DBMS und den SQL Client (pgAdmin 4) möglichst einfach, abgekapselt und unkompliziert auf Ihrem Computer einzurichten, haben wir als Alternative ein auf Docker<sup>7</sup> basiertes Setup vorbereitet. Installieren Sie dazu die Docker Engine<sup>8</sup> und Docker Compose<sup>9</sup>. Kopieren Sie im Anschluss das vorbereitete `docker-composeOLAT.yaml` in einen Ordner Ihrer Wahl und führen Sie es mit `docker-compose` aus. Da es sich dabei um ein Command Line Tool handelt, muss die Ausführung des Befehls im Terminal erfolgen. Wechseln Sie dazu im Terminal in den Ordner mit der heruntergeladenen Datei und führen Sie den Befehl `docker-compose up` aus. Dieser Befehl führt die Instruktionen im `docker-composeOLAT.yaml` aus. Dazu werden zuerst die Docker Images `postgres:13.4` und `dpag/pgadmin4:5.6` heruntergeladen (sofern diese nicht schon auf Ihrem System vorhanden sind). Anschließend werden die zwei Services `db` und `pgadmin` konfiguriert, gestartet und ihre Ports auf ihr lokales System gemapped.

Alternativ können Sie `docker-compose up` auch mit dem zusätzlichen Parameter `-d` im detached Modus ausführen. Um die Container wieder zu beenden, müssen Sie den Befehl `docker-compose down` ausführen.

### Hinweis



Für die meisten Linux Distributionen sollte es ausreichend sein, die Pakete aus den System Repositories zu installieren. Sollten diese nicht vorhanden sein oder es andere Probleme geben (z.B. zu alte Versionen), dann können Sie auch die für ihre Distribution passende Anleitung von [docker docs<sup>a</sup>](#) verwenden.

Für Windows sollten Sie ebenfalls die entsprechende Anleitung auf [docker docs<sup>a</sup>](#) verwenden. Zusätzlich könnte Ihnen auch das Docker Beginner Tutorial 4–6 von Raghav Pal<sup>b</sup> oder eine Google-Suche helfen.

<sup>a</sup><https://docs.docker.com/get-docker/>

<sup>b</sup><https://www.youtube.com/playlist?list=PLhW3qG5bs-L99pQsZ74f-LC-t0EsBp2rK>

Sobald die Services erfolgreich gestartet wurden, können Sie die pgAdmin4 Weboberfläche unter <http://localhost:8080> erreichen. Achtung: Abhängig von ihrem System kann es (speziell beim ersten Start) Minuten dauern, bis die Oberfläche zur Verfügung steht. Als E-Mail Adresse wurde `pgadmin@example.com` vergeben und das Passwort lautet `SuperSecret`. Dort können Sie eine Datenbankverbindung einrichten. Verwenden Sie dafür den Server-Dialog wie in der pgAdmin Dokumentation<sup>10</sup> beschrieben. In der Weboberfläche ist die PostgreSQL Datenbank unter dem Hostname `db` erreichbar. Der Benutzer ist `postgres` und das Passwort lautet `SuperSecret`. Zusätzlich ist der Port des PostgreSQL Service (`db`) auch auf ihren localhost gemapped, Sie können die Datenbank also auch von ihrem System aus (z.B. mit DBeaver) erreichen, indem Sie als Hostname `localhost` verwenden der Benutzer und das Passwort bleiben dabei gleich.

<sup>5</sup><https://pgadmin.org>

<sup>6</sup><https://dbeaver.io>

<sup>7</sup><https://www.docker.com>

<sup>8</sup><https://docs.docker.com/engine>

<sup>9</sup><https://docs.docker.com/compose>

<sup>10</sup><https://www.pgadmin.org/docs/pgadmin4/5.6/connecting.html>

### Hinweis



Sollten Sie Docker Toolbox verwenden, wird Docker in einer virtuellen Maschine (VirtualBox) ausgeführt. Das bedeutet, dass Sie anstelle von localhost die entsprechende IP Adresse verwenden müssen. Führen Sie dafür den Befehl `docker-machine ip default` aus, um die IP Adresse auszulesen.

Sollte die IP beispielsweise 192.168.99.100 sein, können Sie mit <http://192.168.99.100:8080> auf die pgAdmin4 Oberfläche zugreifen.

## Diskussionsteil (im PS zu lösen; keine Abgabe nötig)

- a) ☐ ★ Lösen Sie den Sheet 00 in OLAT.

### Hinweis



Die korrekte Abgabe Ihrer Lösung ist wichtig, damit Ihnen keine Punkte verloren gehen (siehe [information.<sup>OLAT</sup>pdf](#)).

Sollten dabei Probleme auftreten, dann lassen Sie sich bitte von Ihren Kolleginnen und Kollegen oder der Kursleiterin und -leiter helfen.

**Diese Unteraufgabe sollte jeder selbst ausführen, um sicherzustellen, dass jeder in OLAT Abgaben machen kann. Dies sollte einem Punkteverlust vorbeugen, da verspätete oder nicht den Vorgaben entsprechende Abgaben aus dem Hausaufgabenteil nicht akzeptiert werden.**

### Hinweis



Geben Sie nur **UTF-8** encodierte Textdateien ab. Verwenden Sie daher einen modernen Texteditor zum Erstellen dieser Dateien. Geeignet/empfohlene Editoren sind VIM<sup>a</sup>, Emacs<sup>b</sup>, Sublime<sup>c</sup>, Virtual Studio Code<sup>d</sup>, Atom<sup>e</sup>, gedit<sup>f</sup> oder Notepad++<sup>g</sup>.

<sup>a</sup><http://www.vim.org>

<sup>b</sup><https://www.gnu.org/software/emacs/>

<sup>c</sup><https://www.sublimetext.com/>

<sup>d</sup><https://code.visualstudio.com/>

<sup>e</sup><https://atom.io/>

<sup>f</sup><https://wiki.gnome.org/Apps/Gedit>

<sup>g</sup><https://notepad-plus-plus.org/>

- b) ☐ ★ Überlegen und Diskutieren Sie (kurz) ob es Ihnen möglich ist die weiter oben am Zettel erwähnten Tools zu installieren und zu verwenden. Die Einrichtung dieser Tools sollte unter Linux, MacOS und Windows möglich sein, das Setup wurde allerdings nur unter Linux getestet.

Das DBMS, der SQL Client und Docker stehen auf den ZID Rechnern nicht zur Verfügung, daher sollten Sie diese Tools auf ihrem eigenen Computer installieren. Sollte Ihnen das aus irgendeinem Grund nicht möglich sein (z.B. Sie besitzen keinen eigenen Computer) dann klären Sie bitte das weitere Vorgehen mit ihrer Kursleiterin / ihrem Kursleiter ab.

## Hausaufgabenteil (Zuhause zu lösen; Abgabe nötig)

### Aufgabe 1 (Theoriefragen)


[5 Punkte]

In dieser Aufgabe werden Sie einige Theoriefragen beantworten. Formulieren Sie Ihre Antworten bitte in *eigenen Worten* – von den Vorlesungsfolien abgeschriebene Lösungen sind nicht gestattet.

- a) 2 Punkte Nennen und erklären Sie fünf Vorteile, die Datenbanksysteme gegenüber herkömmlichen Dateien haben?

#### Abgabe



 exercise\_1a.txt

#### Lösung



DBMS haben folgende Vorteile gegenüber herkömmlichen Dateien.


1. Sie sind effizienter, um große Datenmengen zu verarbeiten.
2. Sie unterstützen Serialisierung und Transaktionen. Dadurch kann z.B. sichergestellt werden, dass Änderungen ganz oder gar nicht durchgeführt werden.
3. Sie erlauben eine feingranulare Zugriffsverwaltung.
4. Es werden Redundanzen und Inkonsistenzen vermieden.
5. Sie erlauben Integritätsverletzungen zu verhindern.

Weitere Details können Sie in den VO Slides nachlesen.

- b) 1.5 Punkte Nennen und erklären Sie die ACID-Kriterien für Transaktionen in Datenbanken.

#### Abgabe



 exercise\_1b.txt

#### Lösung



Die ACID-Kriterien sind:

- \* Atomicity: Die Änderungen sind atomar somit werden entweder alle oder keine Änderungen einer Transaktion durchgeführt.
- \* Consistency: Von einem korrekten Datenbankzustand führt eine Transaktion wieder in einen korrekten Datenbankzustand.
- \* Isolation: Eine Transaktion wird in einem virtuellen Einbenutzerbetrieb durchgeführt. D.h. erst am Ende einer Transaktion werden die Änderungen für alle sichtbar.
- \* Durability: Am ende der Transaktion ist sichergestellt, dass die Änderungen dauerhaft auf der Festplatte geschrieben sind und nicht nur im Cache liegen.

- c) **1.5 Punkte** Erklären Sie kurz die drei Schichten eines DBMS laut der 3-Schichten-Architektur. Welche Aufgaben haben diese Schichten?

#### Abgabe



exercise\_1c.txt

#### Lösung



Ein DBMS mit 3-Schichten-Architektur hat folgende Schichten:

- \* Physische (interne) Schicht: Hierbei handelt es sich um die Schicht, welche den Inhalt auf dem Sekundärspeicher (z.B. Festplatte, SSD) abgelegt. Dazu gehören z.B. auch die Recordstrukturen und Zugriffspfade wie Indices, Hashtabellen, B-Bäume, ...
- \* Konzeptuelle (logische) Schicht: Diese Schicht stellt den logischen Aufbau der Datenbank dar. Dazu gehört eine einheitliche Darstellung von Daten z.B. in Form von Tabellen. Diese werden durch die DDL verwaltet und verändert die in dieser Schicht implementiert ist.
- \* View (externe) Schicht: Mittels dieser Schicht wird die Datenbank angesprochen und verwendet. Ein View repräsentiert dabei eine spezielle Darstellung/Sicht der Daten, die in der Datenbank gespeichert sind. In dieser Schicht werden die DQL und die DML implementiert, die verwendet werden können, um Daten abzufragen oder zu verändern.

Für weitere Details siehe VO Slides.

## Aufgabe 2 (Einrichten der Datenbank)

**[5 Punkte]**

In dieser Aufgabe sollen Sie das am Anfang dieses Zettels (unter Tools) beschriebene Datenbankmanagementsystem (vorzugsweise auf Ihrem eigenen Computer) einrichten und darauf gegebene Datenbankabfragen mit einem SQL Client ausführen.

- a) **4 Punkte** Setzen Sie PostgreSQL 13.4 auf und führen Sie anschließend folgende Query, die Sie auch in `postgres_query_1.sqlOLAT` finden, aus:

```
1  select *
2  from pg_config
3  where name = 'VERSION';
```

Geben Sie das Ergebnis als TXT-Datei ab.

#### Abgabe



postgres\_query\_1.txt

## Lösung



name	setting
VERSION	PostgreSQL 13.4 (Debian 13.4-1.pgdg100+1)

(1 row)

- b) 1 Punkt Erstellen Sie eine Datenbank mit dem Namen db\_ps\_sheet01 indem Sie die Query in `database_create.sql` ausführen:

```
1 create database db_ps_sheet01;
```

Spieren Sie nun den Dump in `database_dump.sql` ein und führen Sie anschließend die SQL Query in `postgres_query_2.sql` aus:

```
1 select
2     name,
3     subdivision,
4     two_letter,
5     subcountry_name
6 from
7     country
8 join subcountry on
9     country.two_letter = subcountry.country
10 where
11     two_letter = 'AT';
```

Geben Sie das Ergebnis als TXT-Datei ab.

## Hinweis



Um die Datenbank zu füllen, spielen wir die Daten aus einer Datei in eine Datenbank ein. Diese Datei enthält einen sogenannten Datenbankdump (kurz Dump)<sup>a</sup>. Eine Möglichkeit, diesen Dump einzuspielen ist, die Dump-Datei in den selben Ordner wie das `docker-compose.yml` zu kopieren und dort dann den Befehl

```
cat database_dump.sql | docker-compose exec -T db psql -U postgres -d db_ps_sheet01
```

ausführen.

Dabei ist `database_dump.sql` der Name der Datei die den Dump enthält und `db_ps_sheet01` ist der Name der Zieldatenbank in die dieser Dump eingespielt werden soll. Es kann sein, dass diese beiden Argumente (Namen) an die Aufgabenstellung anzupassen sind.

<sup>a</sup><https://de.wikipedia.org/wiki/Datenbankdump>

## Abgabe



postgres\_query\_2.txt

**Lösung**

name	subdivision	two_letter	subcountry_name
Austria	Burgenland	AT	1
Austria	Kärnten	AT	2
Austria	Niederösterreich	AT	3
Austria	Oberösterreich	AT	4
Austria	Salzburg	AT	5
Austria	Steiermark	AT	6
Austria	Tirol	AT	7
Austria	Vorarlberg	AT	8
Austria	Wien	AT	9

(9 rows)

**Wichtig:** Laden Sie bitte Ihre Lösung in OLAT hoch und geben Sie mittels der Ankreuzliste auch unbedingt an, welche Aufgaben Sie gelöst haben. Die Deadline dafür läuft am Vortag des Proseminars um 23:59 (Mitternacht) ab.