

10.10.2023

Übungsblatt 1 – Lösungsvorschlag

Diskussionsgrundlage Proseminar

In diesem Semester werden wir die Proseminareinheiten intensiv für das gemeinsame Diskutieren und Lösen von Beispielen verwenden. Wie in [information.pdf^{OLAT}](#) beschrieben, hat jedes Übungsblatt einen Abschnitt mit kurzen Übungsaufgaben, die als Diskussionsgrundlage für die jeweilige Proseminareinheit dienen. Diese Aufgaben, die wir während der Proseminareinheit lösen, dienen dazu, Sie auf die Hausübungen sowie auf die Vorlesungsklausur vorzubereiten.

Hinweis



Die Proseminareinheiten sollen für Sie ein Forum darstellen. Lösen Sie die während einer Proseminareinheit gestellten Aufgaben in Gruppen und diskutieren Sie verschiedene Herangehensweisen. Gestalten Sie die Proseminareinheiten mit, fragen Sie Ihre*n Proseminarleiter*in, Ihre Kolleg*innen und nutzen Sie die Zeit, um sich intensiv über die Inhalte auszutauschen.

Hinweis



Stellen Sie Fragen, auch wenn diese Ihnen noch so „blöd“ erscheinen – nur so können wir gemeinsam bestehende Unklarheiten beseitigen. 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.

Diagramme

In diesem Proseminar werden wir häufig sogenannte ER-Diagramme zeichnen. Als Editor für Diagramme können wir zum Beispiel [yEd¹](#) empfehlen. Sollten Sie sich für diesen Editor entscheiden, so können Sie auf das Template [ER-master.graphml^{OLAT}](#) zurückgreifen, um schneller einzelne ER-Elemente zu zeichnen. Selbstverständlich können Sie auch gerne auf andere Tools wie [Draw.io²](#) zurückgreifen oder Zeichnungen von Hand anfertigen, die Sie anschließend fotografieren.

¹<https://yworks.com/products/yed>

²<https://about.draw.io>

Datenbankmanagementsystem

Als Datenbankmanagementsystem (DBMS) verwenden wir *PostgreSQL 15*³. Alle Aufgaben wurden auf diesem System erstellt und getestet. Von einer Verwendung von *MySQL* oder *MariaDB*⁴ raten wir für dieses Proseminar ab, da sich die Funktionalität und die SQL-Syntax von *PostgreSQL* und *MySQL* bzw. *MariaDB* leicht unterscheiden. Es steht Ihnen natürlich frei, für die Ausarbeitung Ihrer Lösungen ein beliebiges DBMS zu verwenden, solange Ihre Abgaben mit *PostgreSQL 15* kompatibel sind.

SQL-Client

Als SQL-Client können wir Ihnen *pgAdmin*⁵ empfehlen. Natürlich steht es Ihnen auch hier frei, andere Tools wie zum Beispiel *DBeaver*⁶ oder *Beekeeper Studio*⁷ zu wählen.

Docker (Optional)

Sie können gerne das DBMS 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*⁸ basiertes Setup vorbereitet.

Installieren Sie dazu die *Docker Engine*⁹ und *Docker Compose*¹⁰ auf Ihrem System. Kopieren Sie anschließend `docker-composeOLAT.yml` in einen Ordner Ihrer Wahl und führen Sie `docker-compose` aus. Da es sich hierbei um ein Kommandozeilen-Programm 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 in der Datei `docker-composeOLAT.yml` aus. Dazu werden zuerst die Docker Images `postgres:15.4` und `dpape/pgadmin4:7.6` 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, führen Sie den Befehl `docker-compose down` aus.

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*^a heranziehen.

Für Windows sollten Sie ebenfalls die entsprechende Anleitung auf *docker docs*^a zurückgreifen. Zusätzlich könnte Ihnen auch das „Docker Beginner Tutorial“ 4–6 von Raghav Pal^b oder eine Google-Suche helfen.

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

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

³<https://postgresql.org/docs/15/index.html>

⁴<https://mariadb.org>

⁵<https://pgadmin.org>

⁶<https://dbeaver.io>

⁷<https://github.com/beekeeper-studio/beekeeper-studio>

⁸<https://docker.com>

⁹<https://docs.docker.com/engine>

¹⁰<https://docs.docker.com/compose>

Sobald die Services erfolgreich gestartet wurden, können Sie die *pgAdmin4*-Weboberfläche unter `localhost:8080` erreichen. Dort können Sie eine Datenbankverbindung einrichten.

Hinweis



Abhängig von Ihrem System kann es – speziell beim ersten Start – einige Minuten dauern, bis die Oberfläche zur Verfügung steht.

Als E-Mail-Adresse wurde `pgadmin@example.com` vergeben und das Passwort lautet `SuperSecret`. Verwenden Sie dafür den Server-Dialog wie in der *pgAdmin*-Dokumentation¹¹ 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 der 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 – Benutzer und Passwort bleiben dabei unverändert.

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 das Übungsblatt 0.

Hinweis



Die korrekte Abgabe Ihrer Lösung ist wie in `information.OLATpdf` beschrieben wichtig, damit Ihnen keine Punkte verloren gehen.

Sollten dabei Probleme auftreten, dann bitten Sie bitte Ihre Kolleg*innen oder Ihre*n Proseminarleiter*in um Hilfe.

Diese Unteraufgabe dient dazu die korrekte Abgabe von Lösungen zu üben. Wir empfehlen Ihnen diese Unteraufgabe zu bearbeiten, um Punkteverluste durch inkorrekte Abgaben bei den nachfolgenden Übungsblättern zu vermeiden.

¹¹<https://pgadmin.org/docs/pgadmin4/7.6/connecting.html>

Hinweis



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

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

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

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

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

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

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

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

- b) ☐ ★ Überlegen und Diskutieren Sie (kurz), ob es Ihnen möglich ist, die weiter oben erwähnten Tools auf ihrem eigenen System zu installieren und zu nutzen. Die Einrichtung dieser Tools sollte unter *Linux*, *MacOS* und *Windows* möglich sein – das Setup wurde unter *Linux* und *Windows* getestet.

Hinweis



Das DBMS, der SQL-Client und Docker stehen auf den ZID-Rechnern nicht zur Verfügung, daher sollten Sie diese Tools auf ihrem eigenen System installieren. Sollte Ihnen dies aus irgendeinem Grund nicht möglich sein, klären Sie bitte das weitere Vorgehen mit Ihrem*er Proseminarleiter*in ab.

Hausaufgabenteil (Zuhause zu lösen; Abgabe nötig)

Aufgabe 1 (Theoriefragen)

[5 Punkte]

In dieser Aufgabe werden Sie einige Theoriefragen beantworten. Formulieren Sie Ihre Antworten in *eigenen Worten* – reines von den Vorlesungsfolien abschreiben ist für Sie nicht zielführend und Ihre Abgabe wird in diesem Fall als nicht gelöst gewertet.

- a) ☐ 2 Punkte Nennen und erklären Sie fünf Vorteile, die Datenbankmanagementsysteme (DBMS) 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 finden Sie in den Vorlesungsunterlagen.

- 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) ablegt. 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.

Weitere Details finden Sie in den Vorlesungsunterlagen.

Aufgabe 2 (Einrichten der Datenbank)

[5 Punkte]

In dieser Aufgabe sollen Sie das am Anfang dieses Übungsblatts unter Tools beschriebene Datenbankmanagementsystem – vorzugsweise auf Ihrem eigenen System – einrichten und darauf gegebene Datenbankabfragen mit einem SQL-Client ausführen.

- a) 4 Punkte Richten Sie *PostgreSQL* 15.4 ein und führen Sie anschließend folgende Query, die Sie auch in `postgres_query_1.sql` finden, aus:

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

Geben Sie das Ergebnis als TXT-Datei ab.

Abgabe



postgres_query_1_result.txt

Lösung



name	setting
VERSION	PostgreSQL 15.4 (Debian 15.4-1.pgdg120+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;
```

Spielen Sie nun den Dump in `database_dump.sql` ein und führen Sie anschließend die 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^a. Eine Möglichkeit, diesen Dump einzuspielen ist, die Dump-Datei in denselben Ordner wie das `docker-compose.yml`^{OLAT} zu kopieren und dort folgenden Befehl ausführen:

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

Dabei ist `database_dump.sql` der Name der Datei, die den Dump enthält, und `db_ps_sheet01` der Name der Zieldatenbank in die dieser Dump eingespielt werden soll. Es kann sein, dass Sie diese beiden Namen an die jeweilige Aufgabe anpassen müssen.

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

Abgabe



postgres_query_2_result.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 16:00 ab.