



Handout zum Übungsbetrieb mit **git**

Dieses Semester setzen wir in der Vorlesung Advanced SQL zur

- Bereitstellung von Übungsblättern,
- Abgabe (Upload) eurer Lösungen und
- Verteilung von Korrekturhinweisen

das Versionskontrollsystem **git** ein. Generelle Informationen zu **git** findet ihr im Web auf <http://git-scm.com>. Ihr findet hier alle notwendigen Details, um **git** für den Übungsbetrieb nutzen zu können. Bitte lest die Hinweise aufmerksam—auch Details sind wichtig. Nach einer kurzen Eingewöhnungsphase werden euch die notwendigen Kommandos ganz automatisch von der Hand gehen.

Setup

Die folgenden Schritte sind **während des gesamten Semesters nur einmalig** auszuführen.

1. **Installiert git.** Unter <http://git-scm.com/downloads> findet ihr Installationshinweise für jedes gängige Betriebssystem. Alternativ ist **git** auch bereits im Paket-Manager eures Betriebssystems verfügbar. Wir empfehlen euch, mit **git** auf der Kommandozeile zu arbeiten, genau wie wir auch. Zur Kommandozeilen-Variante können wir euch Hinweise und Hilfestellungen geben. Solltet ihr graphische Varianten einsetzen, so geht das in Ordnung—ihr seid dann aber auf euch allein gestellt.
2. **Erstellt einen Account auf GitHub (<http://github.com>).** Solltet ihr dort bereits einen Account besitzen, könnt ihr diesen auch für den Übungsbetrieb einsetzen.
Wir nutzen den *Classroom*-Service auf *GitHub*, um unsere Aufgabenblätter und eure Lösungen in sog. *Repositories* (kurz: *Repos*) zu speichern. Diese *Repos* sind *nicht* öffentlich zugänglich.
3. **Registriert euch im Forum zur Vorlesung (<https://forum-db.informatik.uni-tuebingen.de/c/ss22-asql>).** Wir werden das Forum zur Diskussion von Vorlesungsinhalten aber auch zur Verwaltung des Übungsbetriebes nutzen. Daher ist die Anmeldung im Forum *nicht* optional.
4. **Öffnet das Forum-Posting “Der Advanced SQL Übung bei GitHub Classroom beitreten”.** Nutzt den Kontakt zu euren Kommilitonen in den ersten Vorlesungen oder auch den entsprechenden Thread im Forum, um ein **Einzel- oder Zweier-Team** für die Übungen zu bilden. Legt einen Namen für das Team fest.
Ihr findet in dem Posting einen Link, der euch auf classroom.github.com weiterleitet. **Tretet dort einem existierenden Team bei oder legt gegebenenfalls ein neues Team an (Create a new team)**, dem dann eure Partnerin/euer Partner später beitreten kann.
5. **Auf GitHub ist nun ein Repo für euer Team angelegt.** Folgt dem Link zu diesem Repo (**Your assignment has been created here:...**). Notiert euch die URL des Repos:

`https://github.com/DBatUTuebingen/asql-ss22-⟨YourTeam⟩`

wobei `⟨YourTeam⟩` aus dem Namen eures Teams abgeleitet wurde.

6. **Erstellt eine lokale Kopie des Repos auf eurem Rechner.** Sucht euch dazu ein lokales Directory, in dem ihr für den Rest des Semesters die Files zum Übungsbetrieb organisieren wollt. In diesem Directory das folgende **git**-Kommando ausführen:

```
1 | git clone https://github.com/DBatUTuebingen/asql-ss22-⟨YourTeam⟩.git
```

Innerhalb des dadurch erzeugten (jetzt noch fast leeren) lokalen Directories namens `asql-ss22-⟨YourTeam⟩` findet eure gesamte Arbeit an den Übungen statt.

7. **Damit euer Repo später auch unsere Übungsblätter aufnehmen kann, bereitet euer Repo darauf wie folgt vor (stellt dabei sicher, dass ihr euch im lokalen Directory `asql-ss22-⟨YourTeam⟩` befindet):**

```
1 | git remote add assignments https://github.com/DBatUTuebingen/asql-ss22.git
```

Das wär's. Wie gesagt: der obige Setup ist einmalig und wiederholt sich während des Semesters nicht.

Download von Übungsblättern

Die folgenden Schritte sind **für jedes Übungsblatt erneut** auszuführen, also normalerweise wöchentlich. Sie gelten auch dann, wenn wir Updates/Korrekturen an einem Übungsblatt vornehmen—darauf weisen wir euch im Forum gegebenenfalls gesondert hin.

1. Wechselt in euer lokales Directory `asql-ss22-<YourTeam>`.
2. Führt das folgende `git`-Kommando aus:

```
1 | git pull assignments master
```

3. Danach findet ihr das Übungsblatt in einem neuen Subdirectory `assignment<week>` vor.

Upload eurer Lösungen

Die folgenden Schritte sind immer dann auszuführen, wenn ihr **eure Lösungen zu einem Übungsblatt abgeben** wollt, also normalerweise wöchentlich. Eine zuvor hochgeladene Lösung könnt durch einen erneuten Upload durch eine neue Version ersetzen, solange der Abgabzeitpunkt noch nicht erreicht ist. Spätere Versionen werden bei der Korrektur nicht berücksichtigt und werden ggf. durch die korrigierte Version ersetzt.

1. Bearbeitet das Übungsblatt und hinterlasst eure Lösungen in (eventuell mehreren) Dateien innerhalb eines neuen Subdirectories `solution<week>`.
2. Wenn eure Lösung komplett ist, führt die folgenden Kommandos aus, um die Lösung unseren Tutoren zugänglich zu machen. Wechselt in das Directory `<YourTeam>` oder in eines der Subdirectories. Zeilen mit `#` sind lediglich Erläuterungen:

```
1 | # Abfrage: Welche Dateien sind neu (bzw. veraendert) und gehoeren zur Loesung?
2 | git status
3 | # Fuer Loesung relevante neue (bzw. modifizierte) Dateien bei git bekanntmachen:
4 | git add <file1> <file2> ...
5 | # Diese Dateien im Repo hinterlegen, selbstsprechenden Kommentar hinzufuegen:
6 | git commit -m "Abgabe zum Uebungsblatt <week>"
7 | # Dateien auf GitHub hochladen:
8 | git push
```

Achtung: Erst nach dem `git push` sind eure Lösungen für die Tutoren zugreifbar.

Solltet ihr Updates an euren Dateien `<file1>`, `<file2>`, ... hochladen wollen, sind `git add <file1> ...`, `git commit -m "..."` und `git push` erneut auszuführen. Sollte eine neue Datei, sagen wir `<file3>`, zur Lösung hinzukommen, ist zuvor auch ein `git add <file3>` notwendig.

Verteilung von Korrekturhinweisen

Das folgende Kommando führt ihr immer dann aus, wenn ihr die Korrekturhinweise und Bewertung eurer Lösungen erhalten wollt. Unsere Tutoren hinterlassen die Korrekturhinweise direkt in (neuen Versionen) eurer Files und eine Punkteübersicht in der Datei `README.md`:

1. Wechselt in das Directory `<YourTeam>` oder in eines der Subdirectories.
2. Empfängt die Korrekturen von *GitHub*:

```
1 | git pull
```

Hinweis: Um uns die Korrektur direkt in eurer Lösung und euch einen sinnvollen Einsatz von `git` bei der Teamarbeit zu ermöglichen, müssen die Files eurer Lösung grundsätzlich im Plain-Text-Format vorliegen. Antworten auf Fragen können dabei einfach als Kommentare in Code-Files verfasst werden, PDF- oder gar Word-Dateien werden **nicht** korrigiert.