

# LU03c - Projekt 1: freeDB

## Einleitung

Der Grundgedanke des Innovationsmodul II ist es eine Innovation vorzubereiten und den Nachweis zu bringen, dass diese grundsätzlich möglich ist. Gemäss Definition ist unter dem Begriff Innovation *eine neue Idee/Produkt oder eine neue, bessere Zusammensetzung von Komponenten oder Prozessen* zu verstehen.

## Ausgangslage

Datenbanken gibt es sehr viele auf dem Markt, die meisten sind bezüglich der Lizenzierung für Privatpersonen kostenlos. Zudem gibt das Thema *Offenlegung des Sourcecodes* immer wieder Grund zur Diskussion bei Unternehmen.

Dieses Projekt soll das Prinzip der Datenbank neu denken und die Themen *Lizenzierung* und *Openbsource*, neu definieren. D.h. das Projekt ist als OpenSource-Projekt durchzuführen, was bedeutet, dass auch Unternehmen keine Lizenzgebühren zahlen müssen.

## Projektauftrag freeDB

*Re-Programmieren* Sie die CRUD-Operationen einer Datenbank unter Verwendung eines Standard-Befehlssatzes einer Programmier- oder Scriptsprache (Python, BASH, ...).

Achten Sie darauf, dass jedes Teammate für einen ASPEKT der CRUD-Operationen zuständig ist. Sprich Person A erledigt das CREATE, eine andere Person das READ, etc.

## Befehlsreferenz C = Create

1. CREATE Table bzw. CREATE TABLE IF NOT EXITSTS
  1. Separate Files anlegen für mit Tabellenstruktur und Tabelleninhalt, erkennbar an der Endung
  2. Falls File vorhanden ist □ entsprechende Fehlermeldungen ausgeben lassen
2. INSERT INTO TABLE
  1. Prüfen, ob Tabellen und genannten Spalten vorhanden sind.
  2. Zeile am Ende des Files dranhängen
  3. Gesamten Fileinhalt Sortierten nach der 1. Spalte (ID)

## Befehlsreferenz R = Read

1. SELECT count(\*)
  1. AWK auf Datenfile
  2. Alternativ wc (Wordcount) verwenden
2. SELECT \* FROM
  1. Fehlermeldung, wenn Tabellen nicht existiert
  2. Gesamter Inhalt der Tabelle wird ausgegeben
3. SELECT nach Spalten
  1. SELECT Spalte1, Spalte2, Spalte3 FROM Tabelle
  2. Wenn Spalte nicht existiert, Fehlermeldung.
4. SELECT mit WHERE
  1. SELECT \* FROM Tab WHERE ID=15
  2. GREP kombiniert mit AWK
  3. Spalte auswählen, die gefiltert werden soll und darauf den GREP anwenden  RegEx
  4. Die betroffene Zeile ausgeben lassen
5. SELECT mit JOIN (zweiter Entitätentyp)
  1. Gemeinsamer Schlüssel zwischen zwei Tabellen
  2. Treffermenge beider Tabellen ausgeben gemeinsam Ausgeben lassen
6. SELECT mit Bedingung/WHERE (z.B. Gehalt > 10'000)
  1. SELECT MIT AWK
  2. Spaltenwert muss Vergleichbar sein
7. SELECT mit LIKE
  1. SELECT MIT AWK und GREP

## Befehlsreferenz U = Update

1. UPDATE TABLE WHERE ID=??
  1. Zeile finden herauslesen
  2. Mittels RegEx Zeile aktualisieren
2. ALTER TABLE
  1. In Tabellendefinition prüfen, ob die Spalte schon vorhanden ist
  2. Wenn nicht, Spalte am Ende anhängen
  3. Datenfile mit Leeren Kommas füllen am Ende auffüllen

## Befehlsreferenz D = Delete

1. DELETE FROM TABLE
  1. AWK mit GREP kombiniert
  2. grep -v suchmuster originaldatei > tempdatei
  3. mv tempdatei originaldatei
2. DROP TABLE
  1. rm-Befehl

## Zusatz

1. INDEXIERUNG (Sortierung der Tabelleninhalte)
  1. Select auf eine String-Spalte, die unsortiert ist (Like «demi%»)
  2. Schattentabellen mit entsprechend Sortiert Spalte anlegen
  3. Gleiche Abfrage auf die Schattentabelle
  4. Zeit vor der Indexierung und Zeit nach der Indexierung stoppen
  5. <https://wiki.ubuntuusers.de/Skripte/Stoppuhr/>
2. Fonetische Suchen via Soundex
  1. Hierbei geht es eine Klang-Suche anstatt nach Wörtern.
  2. SoundEx ermöglicht das Auffinden von Inhalten mit anderen Schreibweisen.
  3. <https://gist.github.com/livibetter/1997957>
  4. <https://de.wikipedia.org/wiki/Soundex>

From:  
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:  
<https://wiki.bzz.ch/modul/m245/learningunits/lu03/theorie/01>

Last update: **2025/11/17 08:33**

