

Datenbanken 1 – PS (501.073)

Projektabgabe – Teil 4

Abzugeben bis **8.6.2018 (22:00)**

Im Folgenden finden Sie das relationale Schema zum ER-Diagramm der letzten Abgabe. Die Primärschlüssel sind unterstrichen, die Fremdschlüsselbeziehungen sind in **rot** markiert.

Ihre Aufgabe besteht nun darin, dieses relationale Schema in SQL umzusetzen, also die Tabellen (mittels CREATE TABLE ... Anweisungen) zu generieren. Erstellen Sie dazu eine Datei create.sql, welche **alle** Anweisungen enthält. *Bitte achten Sie auf den korrekten Dateinamen.* Die create.sql Datei ist die einzige Datei die abgegeben werden soll.

Hinweis: Die create.sql Datei **muss** von PostgreSQL geladen werden können. Zum Testen könnten Sie auf Ihrem eigenen System beispielsweise eine Datenbank mydb anlegen und dann Ihre create.sql Datei laden:

```
createdb mydb
psql -d mydb -f create.sql
```

Wir werden die Abgaben automatisch überprüfen. Zunächst muss Ihr Skript auf PostgreSQL ausführbar sein, sonst wird Ihre Abgabe nicht weiter überprüft. Es gibt jeweils 0.4 Punkte für jede der folgenden Kategorien:

- jede Tabelle existiert mit richtigem Namen
- jedes Attribut existiert mit richtigem Namen und Datentyp
- jeder Primärschlüssel existiert
- jeder Fremdschlüssel existiert
- jede Beschränkung der erlaubten Datenwerte wurde angelegt

Die Kategorien werden in dieser Reihenfolge überprüft. Sobald in einer Kategorie ein Fehler gefunden wurde, bricht die Evaluation ab – diese und alle weiteren Kategorien werden dann mit 0 Punkten gewertet. Ein erneuter Upload ist jederzeit möglich.

Die automatische Evaluierung erfolgt mit leichter Verzögerung – es kann sein, dass Sie einige Minuten warten müssen, bis das Ergebnis vorliegt.

Abbildung von Entitäten

Anmerkung: Alle Fremdschlüsselattribute müssen NOT NULL sein.

Person[Nickname, Firstname, Lastname, Password, Birthdate]

Bitte um folgende Benennung + Datentypen im create.sql Skript: Tabellenname: Person

Nickname	TEXT
Firstname	TEXT
Lastname	TEXT
Password	TEXT
Birthdate	TIMESTAMP

Anmerkung: Die Werte von Password müssen NOT NULL und länger als 0 Zeichen sein (Benutzen Sie dazu die Funktion char_length).

Posting[PostingID, Nickname, Time, Location]

$\pi_{\text{Nickname}}(\text{Posting}) \subseteq \pi_{\text{Nickname}}(\text{Person})$

Bitte um folgende Benennung + Datentypen im create.sql Skript: Tabellenname: Posting

PostingID	BIGINT
Nickname	TEXT
Time	TIMESTAMP
Location	TEXT

Anmerkung: Die Werte von Time müssen NOT NULL sein.

Comment[CommentID, PostPostingID, PostNickname, CommenterNickname, CommentText]

$\pi_{\text{PostPostingID, PostNickname}}(\text{Comment}) \subseteq \pi_{\text{PostingID, Nickname}}(\text{Posting})$

$\pi_{\text{CommenterNickname}}(\text{Comment}) \subseteq \pi_{\text{Nickname}}(\text{Person})$

Bitte um folgende Benennung + Datentypen im create.sql Skript: Tabellenname: Comment

CommentID	BIGINT
PostPostingID	BIGINT
PostNickname	TEXT
CommenterNickname	TEXT
CommentText	TEXT

Information[InformationID, PostingID, Nickname, Size]

$\pi_{\text{PostingID, Nickname}}(\text{Information}) \subseteq \pi_{\text{PostingID, Nickname}}(\text{Posting})$

Bitte um folgende Benennung + Datentypen im create.sql Skript: Tabellenname: Information

InformationID	BIGINT
PostingID	BIGINT
Nickname	TEXT
Size	BIGINT

Data[InformationID, Link]

$\pi_{\text{InformationID}}(\text{Data}) \subseteq \pi_{\text{InformationID}}(\text{Information})$

Bitte um folgende Benennung + Datentypen im create.sql Skript: Tabellenname: Data

Link	TEXT
------	------

Anmerkung: Die Werte von Link müssen NOT NULL und länger als 0 Zeichen sein. Die Datentypen und Namen der anderen Attribute entnehmen Sie den entsprechenden Tabellen.

Text[InformationID, Message]

$\pi_{\text{InformationID}}(\text{Text}) \subseteq \pi_{\text{InformationID}}(\text{Information})$

Bitte um folgende Benennung + Datentypen im create.sql Skript: Tabellenname: Text

Message	TEXT
---------	------

Anmerkung: Die Werte von Message müssen NOT NULL und länger als 0 Zeichen sein. Die Datentypen und Namen der anderen Attribute entnehmen Sie den entsprechenden Tabellen.

Abbildung von Relationen als Entitäten

Follows[FollowerNickname, FollowerNickname]

$\pi_{\text{FollowerNickname}}(\text{Follows}) \subseteq \pi_{\text{Nickname}}(\text{Person})$

$\pi_{\text{FollowerNickname}}(\text{Follows}) \subseteq \pi_{\text{Nickname}}(\text{Person})$

Bitte um folgende Benennung + Datentypen im create.sql Skript: Tabellenname: Follows, Attribute: FollowerNickname und FollowerNickname.

Likes[PostPostingID, PostNickname, LikeeNickname]

$\pi_{\text{PostPostingID, PostNickname}}(\text{Likes}) \subseteq \pi_{\text{PostingID, Nickname}}(\text{Posting})$

$\pi_{\text{LikeeNickname}}(\text{Likes}) \subseteq \pi_{\text{Nickname}}(\text{Person})$

Bitte um folgende Benennung + Datentypen im create.sql Skript: Tabellenname: Likes, Attribute: PostNickname, PostPostingID, LikeeNickname.

Abbildung von mehrwertigen Attributen als Entitäten

Tags[PostingID, Nickname, Tag]

$\pi_{\text{PostingID, Nickname}}(\text{Tags}) \subseteq \pi_{\text{PostingID, Nickname}}(\text{Posting})$

Bitte um folgende Benennung + Datentypen im create.sql Skript: Tabellenname: Tags

Tag TEXT

Anmerkung: Die Werte von Tag müssen NOT NULL und länger als 0 Zeichen sein. Die Datentypen und Namen der anderen Attribute entnehmen Sie den entsprechenden Tabellen.