LU09b: Üben – JOINs im Reiseblog

Ziel: JOINs selbst anwenden (INNER JOIN & WHERE-Variante). Arbeiten Sie direkt auf der DB aus LU08c.

Aufgabe 1 – Posts mit Autor:in

Geben Sie Post-ID, Titel, Autor aus – einmal mit INNER JOIN, einmal mit WHERE-Schreibweise. Nach Post-ID sortieren.

Erwartete Spalten: `post_id, title, author`

Aufgabe 2 – Posts pro Kategorie „Städtereise“

Listen Sie alle Posttitel auf, die zur Kategorie „Städtereise“ gehören (''slug = 'staedtereise' ''). Sortieren Sie nach Titel.

Aufgabe 3 – Posts mit Autor und Kategorien

Geben Sie Titel, Autor, Kategorie aus (eine Zeile pro Kategorie). Nach Titel, dann Kategorie sortieren.

Tipp: Sie brauchen drei Tabellen (`posts, post_category, categories`) – plus *users*, wenn der Autor gezeigt werden soll.

Aufgabe 4 – Suche nach Autorname

Zeigen Sie alle Titel der Posts, die von „Martin Merten“ geschrieben wurden.

Aufgabe 5 (optional) – Zählen ohne GROUP BY vermeiden

Finden Sie alle Kategorien, in denen Caro Steig mindestens einen Post hat. (Nur Kategorienamen ausgeben, ohne doppelte Zeilen.)

Tipp: `SELECT DISTINCT c.name …`

Lösungen (Beispielausschnitte)

A1 – Variante JOIN:

```
SELECT p.post_id, p.title, u.display_name AS author
```

```
FROM posts p
INNER JOIN users u ON p.author_id = u.user_id
ORDER BY p.post_id;
```

A2 – Kategorie „Städtereise“:

```
SELECT p.title
FROM posts p
INNER JOIN post_category pc ON p.post_id = pc.post_id
INNER JOIN categories c ON pc.category_id = c.category_id
WHERE c.slug = 'staedtereise'
ORDER BY p.title;
```

A3 – Titel, Autor, Kategorie:

```
SELECT p.title, u.display_name AS author, c.name AS category
FROM posts p
INNER JOIN users u ON p.author_id = u.user_id
INNER JOIN post_category pc ON p.post_id = pc.post_id
INNER JOIN categories c ON pc.category_id = c.category_id
ORDER BY p.title, c.name;
```

A4 – Posts von Martin:

```
SELECT p.title
FROM posts p
INNER JOIN users u ON p.author_id = u.user_id
WHERE u.display_name = 'Martin Merten'
ORDER BY p.title;
```

A5 (optional) – Kategorien von Caro (DISTINCT):

```
SELECT DISTINCT c.name
FROM categories c
INNER JOIN post_category pc ON c.category_id =
pc.category_id
INNER JOIN posts p ON pc.post_id = p.post_id
INNER JOIN users u ON p.author_id = u.user_id
WHERE u.display_name = 'Caro Steig'
ORDER BY c.name;
```

Kurz-Referenz

INNER JOIN … ON – moderne, klare Schreibweise.

Mehrere Tabellen nach FROM + WHERE – ältere Variante, fachlich gleichwertig.

Join-Pfad denken: Von der Tabelle loslaufen, die Sie anzeigen möchten, dann Schritt für Schritt zu den benötigten Infos verbinden (z. B. `posts → users` oder `posts → post_category → categories`).

DISTINCT hilft gegen Dubletten bei N:M-Abfragen.

ORDER BY erst am Schluss – bestimmt die Ausgabe-Reihenfolge.

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

Permanent link:
https://wiki.bzz.ch/de/modul/m290_guko/learningunits/lu10/aufgaben/a_joins?rev=1793530267

Last update: **2026/11/01 11:51**