LiteDB Kommandos für die Abfrage mit LiteDB Studio

Hier ein kompaktes, druckfertiges Cheatsheet zur LiteDB SQL-ähnlichen Abfragesprache. Einfach kopieren, als PDF/Print ausgeben – und schon hast du es offline zur Hand.

Übersicht der SQL-Befehle

Befehl	Syntax	Kurzbeschreibung
SELECT	`SELECT [TOP n] <Felder> FROM <Collection> [WHERE <Filter>] [GROUP BY <Felder>]`	
`[HAVING] [ORDER BY [ASC	DESC]] [LIMIT n] [OFFSET n]`	Datenabfrage mit Projektion, Filter, Sortierung, Gruppierung und Pagination
INSERT	`INSERT INTO <Collection> (<Feld1>,...) VALUES (<Wert1>,...)` `INSERT INTO <Collection> (<BsonDocument>)`	Neue Dokumente hinzufügen
UPDATE	`UPDATE <Collection> SET <Feld>=<Wert>[, ...] [WHERE <Filter>]`	Felder bestehender Dokumente ändern
DELETE	`DELETE FROM <Collection> [WHERE <Filter>]`	Dokumente löschen (ohne `WHERE`: alle)
RENAME	`RENAME COLLECTION <alt> TO <neu>` `RENAME INDEX <alt> TO <neu> ON <Collection>`	Umbenennen von Collections oder Indizes
DROP	`DROP INDEX <IndexName> ON <Collection>`	Index löschen
EXPLAIN	`EXPLAIN <SELECT	INSERT

Filter- und Ausdruckssyntax (BsonExpression)

- **Vergleichsoperatoren**
  `=` `!=` `>` `<` `>=` `<=`
- **Logische Verkettung**
  `AND`, `OR`, `NOT`
- **Pfadangaben**
  - Dot-Notation: `Address.City`
  - Array-Zugriff: `Tags[0]`, `[Scores].any(x=>x>50)`
- **Array-/Kollektions-Operatoren**
  - `any`, `all`, `contains`
- **Regex**
  - `/{pattern}/i` (z. B. `/^A.*/i`)

Aggregationen & Funktionen

Funktion	Beschreibung	Beispiel
`COUNT(<Feld	*>)`	Anzahl Elemente/Gruppengröße
`SUM(<Feld>)`	Summe numerischer Werte	`SUM(Price)`
`MIN/MAX(<Feld>)`	Kleinster/Größter Wert	`MAX(Date)`
`AVG(<Feld>)`	Durchschnitt	`AVG(Rating)`
`TOLOWER/TOUPPER()`	Text in Klein-/Großbuchstaben	`TOUPPER(Name)`
`TRIM/LENGTH/SUBSTR`	Zeichenketten-Funktionen	`TRIM(Title)`, `LENGTH(Text)`
`DATETIME(x)`	Wandelt Timestamp/String → DateTime	`DATETIME(CreatedAt)`

> **Hinweis:** Zusätzlich unterstützen alle BsonExpression-Funktionen (docs) u. a. `IIF`, `ISNULL`, `ROUND`, `CEIL`, `FLOOR`.

Beispiele

```
-- 1) Einfache Abfrage mit Filter und Sort
SELECT Name, Age
FROM users
WHERE Age >= 18 AND City = 'Berlin'
```

```
ORDER BY Name ASC
LIMIT 10 OFFSET 0;

-- 2) Gruppierung und Having
SELECT City, COUNT(*) AS Einwohner
FROM users
GROUP BY City
HAVING Einwohner > 1000
ORDER BY Einwohner DESC;

-- 3) Einfügen
INSERT INTO products (Name, Price, Tags)
VALUES ('Kaffeemaschine', 79.99, ['küche', 'elektronik']);

-- 4) Update mit Filter
UPDATE products
SET Price = Price * 0.9
WHERE Tags.contains('elektronik');

-- 5) Löschen
DELETE FROM sessions
WHERE LastAccess < DATETIME('2025-01-01');

-- 6) Collection umbenennen
RENAME COLLECTION oldUsers TO customers;
```

Schnellreferenz Schlüsselwörter

```
SELECT, INSERT, UPDATE, DELETE,
FROM, INTO, VALUES,
WHERE, GROUP BY, HAVING,
ORDER BY, LIMIT, OFFSET,
RENAME COLLECTION, RENAME INDEX,
DROP INDEX, EXPLAIN
```

> **Tipp:** Bei sehr vielen Feldern oder dynamisch konfigurierten Sets lieber die `WHEN ... CONTAINS`-Variante in Switch-Statements nutzen, oder direkt `BsonExpression`-Filter anlegen. Damit bleibst du flexibel und wartbar.

Viel Erfolg beim Ausdrucken und Entwickeln!

Revision #1
Created 4 July 2025 11:32:14 by Stefan Mechler
Updated 4 July 2025 11:32:41 by Stefan Mechler