

Filtern nach Datumswerten

In **PostgreSQL** können Datumsfilter mit speziellen Funktionen wie **EXTRACT()**, **DATE_PART()** oder direkten Vergleichsoperatoren umgesetzt werden. Diese Methoden sind nützlich, um bestimmte Teile eines Datums (z. B. Jahr, Monat, Tag) zu filtern oder Zeiträume festzulegen.

1. Filtern nach Datum

PostgreSQL unterstützt das Filtern von Datumsangaben im Format **YYYY-MM-DD** mit der **WHERE -Klausel** und Vergleichsoperatoren.

Beispiel:

Alle Bestellungen vom **10. März 2024**:

```
1 SELECT *
2 FROM bestellungen
3 WHERE bestelldatum = '2024-03-10';
4
```

2. Filtern nach Datumsteilen

Um bestimmte Teile eines Datums (z. B. Monat oder Jahr) zu filtern, wird **EXTRACT()** oder **DATE_PART()** verwendet.

Filtern nach Monat (März):

```
1 SELECT *
2 FROM bestellungen
3 WHERE EXTRACT(MONTH FROM bestelldatum) = 3;
4
```

Filtern nach Jahr (2024):

SQL

```
1 SELECT *
2 FROM bestellungen
3 WHERE EXTRACT(YEAR FROM bestelldatum) = 2024;
4
```

Alternative: `DATE_PART()` :

```
1 SELECT *
2 FROM bestellungen
3 WHERE DATE_PART('month', bestelldatum) = 3;
4
```

Beide Methoden geben numerische Werte zurück und funktionieren ähnlich.

3. Filtern innerhalb eines Zeitraums

PostgreSQL ermöglicht es, Zeiträume mithilfe von `BETWEEN` oder Vergleichsoperatoren zu definieren.

Beispiel:

Bestellungen zwischen dem **1. März 2024** und **31. März 2024**:

```
1 SELECT *
2 FROM bestellungen
3 WHERE bestelldatum BETWEEN '2024-03-01' AND '2024-03-31';
4
```

Alternative mit Vergleichsoperatoren:

```
1 SELECT *
2 FROM bestellungen
3 WHERE bestelldatum >= '2024-03-01'
4     AND bestelldatum < '2024-04-01';
5
```

4. Filtern nach dynamischen Datumswerten

PostgreSQL bietet Funktionen wie **CURRENT_DATE** , **NOW()** und **INTERVAL** , um dynamisch mit dem aktuellen Datum zu arbeiten.

Beispiel:

Alle Bestellungen der letzten 30 Tage:

```
1 SELECT *
2 FROM bestellungen
3 WHERE bestelldatum >= CURRENT_DATE - INTERVAL '30
  days';
4
```

Bestellungen am aktuellen Tag:

```
1 SELECT *
2 FROM bestellungen
3 WHERE bestelldatum = CURRENT_DATE;
4
```

5. Arbeiten mit TIMESTAMP

Falls die Datumswerte Zeitangaben enthalten (Datentyp `TIMESTAMP`), kann man explizit nur das Datum extrahieren oder Zeitbereiche filtern.

Beispiel:

Bestellungen am **10. März 2024**, unabhängig von der Uhrzeit:

```
1 SELECT *
2 FROM bestellungen
3 WHERE bestelldatum::DATE = '2024-03-10';
4
```

Filtern nach einer Uhrzeitspanne:

```
1 SELECT *
```

```
2 FROM bestellungen
3 WHERE bestelldatum BETWEEN '2024-03-10 08:00:00' AND
  '2024-03-10 18:00:00';
4
```

6. Filtern nach Wochentagen

Mit **EXTRACT(DOW FROM datum)** (Day of Week) kann nach bestimmten Wochentagen gefiltert werden (0 = Sonntag, 1 = Montag, ...).

Beispiel:

Bestellungen an einem Montag:

```
1 SELECT *
2 FROM bestellungen
3 WHERE EXTRACT(DOW FROM bestelldatum) = 1;
4
```

Zusammenfassung

- **EXTRACT()** oder **DATE_PART()** zum Filtern nach Datumsteilen.
- **Vergleichsoperatoren** oder **BETWEEN** für Zeiträume.
- Dynamische Filter mit **CURRENT_DATE** oder **INTERVAL** .
- Für **TIMESTAMP** -Werte kann das Datum durch **::DATE** oder spezifische Zeitbereiche berücksichtigt werden.

Durch diese Methoden können in PostgreSQL Datumsfilter präzise und flexibel angewendet werden.