Komplexe WHERE-Klauseln

Komplexe WHERE-Klauseln in PostgreSQL mit Klammern

Die Tabelle `ba_flights_excerpt` gibt Daten zu Flügen wieder. Mit ihr können komplexe **`WHERE` -Klauseln** veranschaulicht werden, die mehrere Bedingungen enthalten. Das Arbeiten mit Klammern hilft dabei, die Logik zu klären und die Priorität von `AND` und `OR` zu steuern.

---

1. Warum Klammern verwenden?

In SQL hat `AND` eine höhere Priorität als `OR` . Ohne Klammern interpretiert PostgreSQL die Bedingungen möglicherweise anders, als beabsichtigt. Klammern stellen sicher, dass die gewünschte Logik korrekt umgesetzt wird.

---

2. Beispiele: Korrekte Nutzung von Klammern

Beispiel 1: Flüge mit mehreren Bedingungen filtern

Man möchte alle Flüge, die entweder:

- **Von „Berlin" starten**, **mehr als 100 Passagiere** haben, **und über 3000 km geflogen sind**,

oder

- **Von „London" starten**.

Abfrage ohne Klammern:

SQL

```sql
SELECT *
FROM ba_flights_excerpt
WHERE departure_city = 'Berlin'
  AND total_passengers > 100
  OR distance_flown > 3000
  AND departure_city = 'London';

```

Diese Abfrage liefert falsche Ergebnisse, da PostgreSQL `AND` -Bedingungen vor `OR` -Bedingungen verarbeitet.

Korrekte Abfrage mit Klammern:

SQL

```sql
SELECT *
FROM ba_flights_excerpt
WHERE (departure_city = 'Berlin'
       AND total_passengers > 100
       AND distance_flown > 3000)
   OR departure_city = 'London';

```

Beispiel 2: Flüge nach Einnahmen und Gepäckgewicht filtern

Man möchte alle Flüge, die entweder:

- **Mehr als 1000€ aus Gepäck-Einnahmen erzielt haben**

oder

- **Über 500 kg Gepäckgewicht transportiert haben** und **einen Status „completed" haben**.

Abfrage mit Klammern:

SQL

```sql
SELECT *
FROM ba_flights_excerpt
WHERE revenue_from_baggage > 1000
   OR (baggage_weight > 500
       AND status = 'completed');

```

Die Klammern stellen hier sicher, dass die Bedingungen zur Gepäckmenge und dem Status nur gemeinsam geprüft werden.

Beispiel 3: Dynamische Abfrage mit Datum

Man möchte Flüge filtern, die:

- **Nach dem 1. Januar 2024 stattgefunden haben**,
- **Von „Paris" starten** oder **„New York" als Zielstadt haben**.

Abfrage mit Klammern:

SQL

```sql
SELECT *
FROM ba_flights_excerpt
WHERE actual_flight_date > '2024-01-01'
  AND (departure_city = 'Paris'
       OR arrival_city = 'New York');

```

Ohne Klammern würde PostgreSQL die Bedingungen falsch gruppieren, was zu fehlerhaften Ergebnissen führen könnte.

---

3. Regeln für Klammern in komplexen Bedingungen

1. **AND und OR immer klar gruppieren:**

Klammern verdeutlichen, wie Bedingungen zusammengehören.

- Beispiel: `(A AND B) OR C` bedeutet, dass **A und B zusammen gelten müssen**, bevor C geprüft wird.

2. **Bedingungen hierarchisch aufbauen:**

Bei mehreren Bedingungen ist es wichtig, die Struktur der Abfrage leserlich zu halten.

- Nutze Einrückungen, um die Logik zu verdeutlichen.

3. **Kombiniere logische Filter sinnvoll:**

Verwende `IN`, `BETWEEN` oder andere Operatoren, um Abfragen lesbarer zu machen.

---

4. Häufige Operatoren in WHERE-Klauseln

PostgreSQL bietet viele Werkzeuge, um Bedingungen in `WHERE`-Klauseln präzise zu definieren. Hier eine Übersicht:

	Operator	Beschreibung	Beispiel
1	`=`	Gleichheit	`departure_city = 'Berlin'`
2	`<>` oder `!=`	Ungleichheit	`status != 'cancelled'`
3	`<`, `>`, `<=`, `>=`	Größer, kleiner, größer-gleich, kleiner-gleich	`total_passengers > 100`
4	`BETWEEN`	Bereichsprüfung (inklusive Grenzen)	`distance_flown BETWEEN 1000 AND 3000`
5	`IN`	Prüft auf eine Liste von Werten	`departure_city IN ('Berlin', 'Paris')`
6	`LIKE`	Mustervergleich mit Wildcards	`status LIKE 'comp%'`
7	`IS NULL`	Prüft auf Nullwerte	`revenue_from_baggage IS NULL`

+ Neu

---

5. Zusammenfassung

- **Klammern sind essenziell**, um die Logik komplexer Abfragen mit `AND` und `OR` korrekt zu strukturieren.

- Ohne Klammern interpretiert PostgreSQL die Priorität automatisch, was zu unerwarteten Ergebnissen führen kann.
- Klar strukturierte Abfragen sind leichter zu lesen, zu debuggen und zu warten.

Ein gut formatiertes Beispiel zur Veranschaulichung:

SQL

```sql
SELECT flight_id, airline, total_passengers
FROM ba_flights_excerpt
WHERE (departure_city = 'Berlin' AND total_passengers > 150)
   OR (arrival_city = 'Paris' AND revenue_from_baggage > 2000);

```