

# Relationele Gegevensbanken: FAQ & Tips

---

## Reeks 4: Aggregatie, Groepering en HAVING-Clause

### Algemene Concepten

- **HAVING vs. WHERE:**
  - **HAVING:** Filtert rijen op basis van geaggregeerde waarden en kan alleen na aggregatie worden gebruikt.
  - **WHERE:** Filtert rijen vóór aggregatie.
- PostgreSQL ondersteunt geen geneste aggregatiefuncties zoals `MAX(COUNT(...))`. Gebruik in plaats daarvan subqueries.
- Los oefeningen op verschillende manieren op om je SQL-vaardigheden te verbeteren (bijvoorbeeld met JOINS, subqueries en aggregatiefuncties).

### Tips voor Oefeningen

- **Oefening 2:** Begrijp het verschil tussen `DISTINCT COUNT(...)` (aantal unieke groepen) en `COUNT(DISTINCT ...)` (unieke waarden binnen een kolom).
- **Oefening 3:** Gebruik een geschikte aggregatiefunctie, zoals `MIN` of `MAX`, om geboortedata te analyseren.
- **Oefening 7:**
  1. **Eerste subquery:** Groepeer data en bereken een aggregaat (bijv. totale lengte van een playlist).
  2. **Tweede subquery:** Pas een aggregatiefunctie toe op de resultaten van de eerste subquery (bijv. het maximum aantal nummers).
  3. **Outer query:** Verbind data met de berekende aggregaatwaarden.
    - Houd rekening met `ex aequo` (meerdere rijen kunnen dezelfde waarde hebben).
- **Oefening 10:** Voeg artiesten toe zonder 'Pop'-liedjes. Zorg dat `COUNT` gelijk is aan 0 (niet NULL).
- **Oefening 11:** Cast telwaarden naar `numeric` bij het berekenen van gemiddelden om foutieve typen te voorkomen.

---

## Reeks 5: Manipuleren van Datum en Tijd

### Algemene Concepten

- **Datumbewerkingen:**
  - Gebruik `to_char` om timestamps te formatteren, zoals `to_char(starttime, 'DD/MM/YYYY')`.
  - Gebruik functies zoals `to_date` en `to_timestamp` om strings om te zetten naar specifieke datatypes.
- **Tijdsverschillen:**
  - Bereken tijdsverschillen met `extract(epoch from ...)` of `age(...)`.
  - Houd rekening met zomer- en wintertijd, bijvoorbeeld bij `timestamp2 - timestamp1`.

### Tips voor Oefeningen

- **Oefening 3:** Gebruik dynamische functies in plaats van vaste jaartallen ('magic numbers').
  - **Oefening 6:** Controleer of een datum een maandag is:
    - `extract(dow from timestamp)` (0 = zondag).
    - `to_char(timestamp, 'day')` (LIKE 'monday%').
  - **Oefening 9:** Bereken de duur van een uitzending in minuten met `(extract(epoch from endtime) - extract(epoch from starttime)) / 60`.
  - **Oefening 17:** Bereken per genre het gemiddelde tijdstip waarop een track wordt gespeeld, afgerond op uren.
- 

## Reeks 6: Geavanceerde Concepten

### Algemene Concepten

- **Escapen van speciale karakters:**
  - Voor strings met enkele aanhalingstekens, gebruik dubbele aanhalingstekens: `'What ' 's up? '`.
- **WINDOW-functies:**
  - Niet toegestaan in de WHERE-clause. Gebruik subqueries om gefilterde data te selecteren.

### Tips voor Oefeningen

- **Oefening 7:** Controleer bij het bepalen van de top x of alle rijen met gelijke waarden correct worden opgenomen.
  - **Oefening 10:** Combineer meerdere subqueries om rijen met WINDOW-functies correct te filteren.
- 

## Extra Toevoegingen en Algemene Tips

1. **Leesbaarheid:**
    - Gebruik aliassen voor subqueries in de FROM-clause om je code overzichtelijk te houden.
    - Schrijf commentaar bij complexe queries.
  2. **Optimalisatie:**
    - Minimaliseer het aantal subqueries om prestaties te verbeteren.
    - Voeg indexes toe aan kolommen die vaak in WHERE- of JOIN-clausules worden gebruikt.
  3. **Debugging:**
    - Test subqueries afzonderlijk om fouten sneller te identificeren.
    - Controleer of NULL-waarden niet onverwachte resultaten veroorzaken.
  4. **PostgreSQL-documentatie:**
    - Raadpleeg de officiële documentatie voor functies zoals `extract`, `to_char`, en `age`.
- 

Met deze samenvatting ben je goed voorbereid om de concepten en oefeningen te begrijpen en toe te passen!