

DATABASE MUSEUM

Een museum heeft een eenvoudige database waarin ze bijhouden wie een abonnement heeft of heeft gehad en hoe vaak deze mensen in het museum komen. Van alle personen is de naam en woonplaats bekend en in welk kalenderjaar ze een abonnement voor het museum hadden. Elke keer dat een abonnementshouder in het museum komt, wordt hun lidmaatschapskaart gescand en geregistreerd op welke dag een lid het museum heeft bezocht.

Gebruik de *querier* (sql.b207.nl); log in met *leerling* en *dbB304*) om de volgende vragen te beantwoorden. Tenzij anders vermeld is het de bedoeling dat je een query maakt. De query is je antwoord.

VRAGEN BIJ §1.1 BASIS SQL

1. Selecteer alle records uit de tabel *bezoeken* waarin een bezoek in 2018 staat geregistreerd.
2. Selecteer (alleen) de namen van personen uit Groningen.

VRAGEN BIJ §1.2 LOGISCH ZOEKEN

3. Selecteer alle personen uit Zuidhorn met een id kleiner dan 5.
4. Selecteer alle personen die niet uit Zuidhorn komen met een id die 5 of hoger is.
5. Vergelijk de volgende twee query's:

```
SELECT * FROM personen WHERE (woonplaats='Groningen') OR (woonplaats='Roden' AND id < 5)
```

```
SELECT * FROM personen WHERE (woonplaats='Groningen' OR woonplaats='Roden') AND (id < 5)
```

 - a. Geef (zonder eerst te proberen!) van beide query's een algemene beschrijving in woorden.
 - b. Bekijk het resultaat van beide query's en bespreek op basis van de gevonden records concreet het verschil tussen de beide resultaten.

VRAGEN BIJ §1.3 ORDENEN

6. Selecteer de naam en de woonplaats van alle personen, gesorteerd op woonplaats.
7. Het museum wil graag weten hoeveel unieke bezoekers het trekt, ofwel: welke verschillende mensen bezoeken het museum. Daarbij telt het dus niet als iemand vaker komt.
Selecteer de unieke bezoekers in 2019. Geef een overzicht van de bijbehorende id's van de personen.
8. Selecteer alle records van personen met een naam die eindigt op de letter -e.

VRAGEN BIJ §1.4 REKENEN EN GROEPEREN

9. Maak een query die als enige resultaat het totaal aantal kaarthouders in de database weergeeft. Zorg dat de resultaat tabel als kolomnaam *aantal* heeft.
10. Geef een tabel met het aantal bezoekers per jaar (voor alle jaren uit de database).
11. Dezelfde query als de vorige vraag, met als toevoeging: zorg dat het aantal van hoog naar laag gesorteerd staat.
12. Dezelfde query als de vorige vraag, maar nu met het aantal unieke bezoekers.

VRAGEN BIJ §1.5 SUBQUERY'S

13. Wie was de allereerste kaarthouder die het museum bezocht? Geef het persoonsnummer en de datum. HINT: gebruik een subquery waarbij je de minimumwaarde van de datum zoekt.

VRAGEN BIJ EXTRA §1.6 HAVING & IN

14. Welke kaarthouders hadden in het openingsjaar 2018 nog geen abonnement (maar later wel)?
15. Geef een overzicht van de plaatsen met meer dan één kaarthouder met daarachter het aantal kaarthouders in die plaats.
16. Geef het jaar waarin het aantal kaarthouders het kleinst was met daarbij dit aantal.
17. Geef het jaar met het grootste aantal unieke bezoekers met daarbij dit aantal.

ANTWOORDEN

1.

```
SELECT *  
FROM bezoek  
WHERE jaar=2018
```
2.

```
SELECT naam  
FROM personen  
WHERE woonplaats='Groningen'
```
3.

```
SELECT *  
FROM personen  
WHERE woonplaats='Zuidhorn' AND id < 5
```
4.

```
SELECT *  
FROM personen  
WHERE NOT woonplaats='Zuidhorn' AND id >= 5
```
5. Twee query's:
 - a. De eerste selecteert *alle mensen die uit Groningen komen en mensen uit Roden met een id kleiner dan 5*. De tweede selecteert alle mensen die uit Groningen of Roden komen waarvoor geldt dat ze een id kleiner dan 5 hebben.
 - b. Bij de tweede query valt het record van *Evelien* er af, omdat zij id = 5 heeft. Bij de eerste query was het voldoende dat ze uit Groningen kwam, bij de tweede query niet.
6.

```
SELECT naam,woonplaats  
FROM personen  
ORDER BY woonplaats
```
7.

```
SELECT DISTINCT persoon  
FROM bezoek  
WHERE jaar=2019
```
8.

```
SELECT *  
FROM personen  
WHERE naam LIKE '%e'
```
9.

```
SELECT COUNT(*) AS aantal  
FROM personen
```
10.

```
SELECT jaar,COUNT(*) AS aantal  
FROM bezoek  
GROUP BY jaar
```
11.

```
SELECT jaar,COUNT(*) AS aantal  
FROM bezoek  
GROUP BY jaar  
ORDER BY aantal DESC
```
12.

```
SELECT jaar,COUNT(DISTINCT persoon) AS aantal  
FROM bezoek  
GROUP BY jaar  
ORDER BY aantal DESC
```
13.

```
SELECT persoon,datum  
FROM bezoek  
WHERE datum=  
(  
    SELECT MIN(datum)  
    FROM bezoek  
)
```

14. SELECT DISTINCT persoon
FROM kaarten
WHERE persoon NOT IN
(
SELECT persoon
FROM kaarten
WHERE jaar=2018
)
15. SELECT woonplaats,COUNT(woonplaats) AS aantal
FROM personen
GROUP BY woonplaats
HAVING aantal > 1
16. **LET OP:** er zijn twee jaren die allebei een minimum aantal kaarthouders opleveren!
SELECT jaar,COUNT(persoon) AS aantal
FROM kaarten
GROUP BY jaar
HAVING aantal <= ALL
(
SELECT COUNT(persoon) AS aantal
FROM kaarten
GROUP BY jaar
)
17. SELECT jaar,COUNT(DISTINCT persoon) AS aantal
FROM bezoek
GROUP BY jaar
HAVING aantal >= ALL
(
SELECT COUNT(DISTINCT persoon) AS aantal
FROM bezoek
GROUP BY jaar
)