



opgaven

■ Stel volgende vragen in de relationele algebra:

1. geef de namen van alle bedienden die wonen in Eindhoven
2. geef de namen van alle bedienden die niet in Eindhoven wonen
3. geef de namen van alle bedienden die zichzelf als manager hebben
4. geef de naam van de managers met een salaris van meer dan 100.000
5. geef de naam van de bedienden met een manager met een salaris van meer dan 100.000



Opmerkingen:

Bij 3: omdat namen uniek zijn en managers ook bedienden zijn is het niet mogelijk dat een manager een naamgenoot is die niet de bediende zelf is.

1. $\Pi_{person_name}(\sigma_{city = \text{“Eindhoven”}}(employee))$
2. $\Pi_{person_name}(employee) - \Pi_{person_name}(\sigma_{city = \text{“Eindhoven”}}(employee))$
3. $\Pi_{person_name}(\sigma_{person_name = manager_name}(manages))$
4. $\Pi_{manager_name}(\sigma_{salary > 100.000 \wedge person_name = manager_name}(works \times \Pi_{manager_name}(manages)))$

Merk op dat de projectie aan het eind is gebruikt om in het cartesisch product maar 1 keer het attribuut *person_name* te hebben. Je kan de projectie ook weglaten maar dan moet je hernoeming gaan gebruiken. We doen dit bij vraag 5 omdat we daar die *person_name* juist nodig hebben.

5. $\Pi_{manages.person_name}(\sigma_{works.salary > 100.000 \wedge works.person_name = manages.manager_name}(works \times manages))$



opgaven

- Stel volgende vragen in de relationele algebra:
 6. geef de namen van de bedienden die meer verdienen dan hun manager
 7. geef de naam van de bedrijven die gevestigd zijn in een stad waar nog een ander bedrijf gevestigd is
 8. geef de namen van alle bedienden die wonen in de stad waar ze werken
 9. geef de namen van alle bedienden die wonen in een andere stad dan hun manager
 10. geef de naam van de bedrijven die gevestigd zijn in een stad waar geen enkele bediende van dat bedrijf woont



6. $\Pi_{works.person_name}(\sigma_{works.salary > m.salary \wedge works.person_name = manages.person_name \wedge m.person_name = manages.manager_name}(works \times manages \times \rho_m(works)))$
7. $\Pi_{company.company_name}(\sigma_{company.company_name \neq c.company_name \wedge company.city = c.city}(company \times \rho_c(company)))$
8. $\Pi_{works.person_name}(\sigma_{employee.person_name = works.person_name \wedge works.company_name = company.company_name \wedge employee.city = company.city}(employee \times works \times company))$
9. $\Pi_{works.person_name}(\sigma_{employee.city \neq m.city \wedge employee.person_name = manages.person_name \wedge m.person_name = manages.manager_name}(employee \times manages \times \rho_m(employee)))$
10. $company - \Pi_{works.company_name, employee.city}(\sigma_{employee.person_name = works.person_name}(employee \times works))$

Merk op dat dit heel sloppy is want we moeten eigenlijk nog opnieuw omnoemen om exact dezelfde attribuutnamen te verkrijgen, maar dat laten we gewoonlijk weg.



opgaven

■ Stel volgende vragen in de relationele algebra:

11. geef de naam van bedienden wiens manager voor een ander bedrijf werkt dan zij zelf
12. geef de namen van de bedrijven die werknemers hebben die in Eindhoven wonen
13. geef de namen van de bedrijven die geen werknemers hebben die in Eindhoven wonen
14. geef de naam van de bediende met het hoogste salaris
15. geef de naam van de manager met het hoogste salaris



Opmerkingen:

Bij 14: dit is bijna letterlijk 1 van de voorbeelden, namelijk dat met de truuk om de “hoogste” te vinden.

Bij 15: dit is 14, uitgebreid met een stuk om de manager te nemen. Hier zit een addertje onder het gras: als je eerst de bediende met het hoogste salaris neemt en dan checkt of het een manager is zit je fout. Je moet eerst de works tabel beperken (met selectie) tot de managers, en dan die met het hoogste salaris zoeken.



opgaven

■ Wat betekenen de volgende vragen?

1. $\Pi_{city}(company) - \Pi_{city}(employee)$

2. $\Pi_{company.city}(company) - \Pi_{company.city}(\sigma_{employee.person_name = works.person_name \wedge works.company_name = company.company_name \wedge employee.city = company.city}(employee \times works \times company))$

3. $\Pi_{m.person_name}(\sigma_{works.salary < m.salary \wedge works.person_name = manages.person_name \wedge m.person_name = manages.manager_name}(works \times manages \times \rho_m(works)))$

1. Geef de steden waar een bedrijf gevestigd is maar geen bedienden wonen. (De truuk hier is om goed te begrijpen wat er wordt bedoeld en dat dan in *eenvoudig* Nederlands op te schrijven, dus niet zoiets als “Geef al de steden waar een bedrijf gevestigd is behalve de steden waar bedienden van eender welk bedrijf wonen”.)
2. Geef de steden waar een bedrijf gevestigd is en waar niemand woont die voor dat bedrijf werkt. (Merk hier op dat we gebruik maken van de associativiteit van het cartesisch product om geen haakjes te schrijven. Merk ook op dat we een beetje sloppy omgaan met de hernoeming, maar niet zo sloppy dat we bij de tweede projectie “city” zouden schrijven in plaats van “company.city” want dan weten we niet welk city-attribuut er wordt bedoeld. Wij zien wel dat er een eis is dat *employee.city* en *company.city* gelijk zijn, maar het gaat te ver om dit te gebruiken als argument om maar sloppy “city” te schrijven) in plaats van “company.city”.
3. Geef de namen van managers die meer verdienen dan één van hun bedienden.