

Reeks I: basisfunctionaliteit

1. Geef garageid, naam, adres, postcode en woonplaats van alle garages.

```
SELECT * FROM garage
```

2. Geef modelid, model en jaar van alle modellen van het merk 'Audi'.

```
SELECT modelID, modelnaam, jaar FROM model
WHERE merk = 'Audi'
```

3. Geef alle informatie van de eigenaars waarvan de naam begint met een 'V', en sorteer het resultaat op woonplaats en op naam.

```
SELECT * FROM eigenaar
WHERE naam LIKE 'V%'
ORDER BY woonplaats, naam
```

4. Geef de naam, in hoofdletters, van de eigenaars die niet in postcode 9000 wonen.

```
SELECT UPPER(naam) FROM eigenaar WHERE postcode != '9000'
```

5. Geef een lijst van de bestaande modellen als volgt: eerst de eerste twee karakters van het merk, dan een punt en daarachter de modelnaam (bv. 'Vo.Polo' of 'Au.A3').
Gebruik SUBSTR om een deel uit een string te halen en CONCAT of || om strings te concateneren.

```
SELECT SUBSTR(merk,1,2) || '.' || modelnaam FROM model
```

6. Geef alles van de motors met brandstof 'diesel' en een vermogen van meer dan 75KW.

```
SELECT * FROM motor
WHERE brandstof = 'diesel' AND vermogen > 75
```

7. Geef een lijst van interventies, geordend op datum (nieuwste eerst), die uitgevoerd door mechanici Filip of mechanici Emmanuel en waarbij de wagen een kilometerstand had van meer dan 30000 km.
Maak deze oefening 3 maal, 1 maal met OR, 1 maal met ANY en 1 maal met IN.

```
SELECT * FROM interventie
WHERE kmstand > 30000
AND (mechanicien = 'Filip' OR mechanicien = 'Emmanuel')
ORDER BY datum DESC
```

Of

```
SELECT * FROM interventie
WHERE kmstand > 30000
AND mechanicien = ANY('Filip','Emmanuel')
ORDER BY datum DESC
```

Of

```
SELECT * FROM interventie
WHERE kmstand > 30000
AND mechanicien IN ('Filip','Emmanuel')
ORDER BY datum DESC
```

8. Geef van alle wagens het chassisnummer en een prijsindicatie. Gebruik als prijsindicatie 'goedkoop' voor wagens met aankoopprijs kleiner dan 8000, 'gemiddeld' voor wagens met aankoopprijs tussen 8000 en 15000 en 'duur' voor wagens met aankoopprijs groter of gelijk aan 15000. *Gebruik een CASE instructie.*

```
SELECT chassisnr,  
       CASE WHEN aankoopprijs < 8000 THEN 'goedkoop'  
            WHEN aankoopprijs < 15000 THEN 'gemiddeld'  
            ELSE 'duur' END AS prijsindicatie  
FROM wagen
```

9. Geef voor de eigenaars uit de provincies Oost en West-Vlaanderen de naam, het volledige adres en een aanduiding van de provincie ('OVL' of 'WVL'). Baseer u hiervoor op de postcodes. *Gebruik de DECODE functie.*

```
SELECT naam, adres, postcode, woonplaats,  
       DECODE(SubStr(postcode,1,1), '8', 'WVL', '9', 'OVL')  
       AS provincie  
FROM eigenaar  
WHERE SubStr(postcode, 1,1) IN ('8','9')
```

10. Geef de wagenid's van de wagens die al een interventie van het type 'herstelling' hebben ondergaan.

```
SELECT DISTINCT wagenid FROM interventie  
WHERE type = 'herstelling'
```

11. Geef alle informatie van de motoren, maar geef het vermogen weer als 'x KW (= y pk)'. De omrekenfactor van KW naar pk is: 1pk = 1.341 KW.

```
SELECT motorid, modelid, type,  
       vermogen || ' KW (= ' || vermogen*1.341 || ' pk)'  
       AS vermogen  
FROM motor
```

12. Geef de interventies waar de omschrijving het woord 'rem' bevat (onafhankelijk van hoofd of kleine letters).

```
SELECT * FROM interventie  
WHERE UPPER(omschrijving) LIKE '%REM%'
```

13. Geef de wagens met even nummerplaat.

Veronderstel dat het laatste karakter van de nummerplaat een getal is.

```
SELECT * FROM wagen  
WHERE MOD(SUBSTR(nummerplaat, LENGTH(nummerplaat), 1), 2) = 0
```

14. Geef van de wagens het wagenid en de prijs afgerond op de duizendtallen.

```
SELECT wagenid,  
       ROUND(aankoopprijs, -3) AS "afgeronde prijs"  
FROM wagen
```

Reeks II: werken met tijd

Zorg ervoor dat de queries onafhankelijk zijn van de datum waarop je ze maakt.

1. Geef het chassisnr, de nummerplaat en de datum van ingebruikname van de wagens die maximaal zes maanden oud zijn. Geef de data in het formaat '19-06-2003'.

Doe dit met behulp van de functie ADD_MONTHS of met behulp van een INTERVAL.

```
SELECT wagenid, chassisnr, nummerplaat,  
       TO_CHAR(ingebruikname, 'dd-mm-yyyy') as ingebruikname  
FROM wagen  
WHERE ingebruikname >= ADD_MONTHS(CURRENT_DATE, -6)
```

Of

```
SELECT wagenid, chassisnr, nummerplaat,  
       TO_CHAR(ingebruikname, 'dd-mm-yyyy') as ingebruikname  
FROM wagen  
WHERE ingebruikname  
      >= (CURRENT_DATE - TO_YMINTERVAL('00-06'))
```

2. Selecteer de interventies sedert het begin van het (kalender)jaar.

Gebruik de functie EXTRACT of TRUNC.

```
SELECT * FROM interventie  
WHERE EXTRACT(YEAR FROM CURRENT_DATE)  
      = EXTRACT(YEAR FROM datum)
```

Of

```
SELECT * FROM interventie  
WHERE TRUNC(CURRENT_DATE, 'YEAR') = TRUNC(datum, 'YEAR')
```

3. Geef een lijst van interventies die tussen 20 maart 2003 en 20 mei 2003.

```
SELECT * FROM interventie  
WHERE datum BETWEEN TO_DATE('20-MAR-2003')  
                  AND TO_DATE('20-MAY-2003')
```

4. Geef de interventies van de afgelopen 60 dagen.

Tip: Een getal dat wordt opgeteld bij of afgetrokken van een datum wordt geïnterpreteerd als een aantal dagen.

```
SELECT * FROM interventie  
WHERE datum >= CURRENT_DATE - TO_DSINTERVAL('60 00:00:00')
```

Of

```
SELECT * FROM interventie WHERE datum >= CURRENT_DATE - 60
```

Reeks III: gebruik van meerdere tabellen

1. Geef voor elke wagen de wagenid, de nummerplaat en de naam van de eigenaar.

```
SELECT wagenid, nummerplaat, naam AS eigenaar
FROM wagen INNER JOIN eigenaar USING (eigenaarid)
```

2. Geef een lijst van mechaniciens die interventies hebben uitgevoerd voor garage 'All Cars'.

```
SELECT DISTINCT mechaniciens
FROM interventie INNER JOIN garage USING (garageid)
WHERE garage.naam = 'All Cars'
```

3. Geef een lijst van eigenaars (naam) en de garage(s) (naam) waar ze al op bezoek zijn geweest.

Noot: verschillende eigenaars kunnen dezelfde naam hebben.

```
SELECT DISTINCT e.eigenaarid, e.naam eigenaar, g.naam garage
FROM eigenaar e INNER JOIN wagen w
      ON e.eigenaarid = w.eigenaarid
      INNER JOIN interventie i ON w.wagenid = i.wagenid
      INNER JOIN garage g ON i.garageid = g.garageid
ORDER BY e.naam, g.naam
```

4. Maak een lijst van nummerplaten met daarbij het merk, type en bouwjaar van de wagen, de brandstofsoort en het vermogen van de motor.

```
SELECT nummerplaat, merk, type, jaar, vermogen, brandstof
FROM wagen INNER JOIN model using(modelid)
      INNER JOIN motor USING (modelid, motorid)
```

5. Geef de modellen waarvoor geen wagen is terug te vinden in de databank.

```
SELECT model.*
FROM model LEFT JOIN wagen ON model.modelid = wagen.modelid
WHERE wagen.wagenid IS NULL
```

6. Geef een lijst met *alle* eigenaars (naam). Indien er een garage is met dezelfde woonplaats of dezelfde postcode geef dan ook de naam en het volledige adres van deze garage.

```
SELECT e.eigenaarid, e.naam eigenaar, g.postcode p2,
      g.woonplaats g2, g.naam garage
FROM eigenaar e LEFT OUTER JOIN garage g
      ON (e.postcode = g.postcode
      OR e.woonplaats = g.woonplaats)
ORDER BY eigenaarid
```

7. Geef een lijst van wagens die ooit in een tijdspanne van één jaar twee maal een interventie hadden. Geef van beide interventies de datum.

Sommige interventies kunnen meer dan eens voorkomen.

```
SELECT i1.wagenid, i1.datum eerste, i2.datum tweede
FROM interventie i1 INNER JOIN interventie i2 ON i1.wagenid = i2.wagenid
WHERE i2.datum > i1.datum AND i2.datum < ADD_MONTHS(i1.datum, 12)
```

Reeks IV: subqueries

1. Geef een lijst van wagens die niet op interventie zijn geweest in 2004.
Doe dit een keer met behulp van een outer join en een keer met een EXISTS conditie.

```
SELECT wagen.*
FROM wagen LEFT JOIN (SELECT * FROM interventie
                      WHERE EXTRACT(YEAR FROM datum) = 2004)
                ON wagen.wagenid = i.wagenid
WHERE interventieid IS NULL
of
SELECT wagen.* FROM wagen
WHERE NOT EXISTS (SELECT * FROM interventie
                  WHERE EXTRACT(YEAR FROM datum) = 2004
                  AND wagen.wagenid = wagenid)
```

2. Geef per model (merk, modelnaam, jaar) de motor met het grootste vermogen.

```
SELECT model.modelid, merk, modelnaam, jaar, type
FROM model INNER JOIN motor ON model.modelid = motor.modelid
WHERE vermogen >= ALL (SELECT vermogen FROM motor
                      WHERE modelid = model.modelid)
ORDER BY modelid
```

3. Geef een lijst van modellen waarvoor een type motor verkrijgbaar is/was dat eveneens verkrijgbaar was voor de Audi A4 2001.

```
SELECT DISTINCT model.*
FROM model INNER JOIN motor ON model.modelid = motor.modelid
WHERE type IN (SELECT type FROM motor
               WHERE modelid = (SELECT modelid FROM model
                                WHERE merk = 'Audi' AND modelnaam = 'A4'
                                AND jaar = 2001))
AND NOT (modelnaam = 'A4' AND jaar = 2001)
```

4. Geef een lijst van wagens die reeds werden behandeld door alle mechaniciens die werken in Garage De Leeuw.

```
SELECT wagen.* FROM wagen
WHERE NOT EXISTS (SELECT * FROM interventie i1
                  WHERE garageid = (SELECT garageid FROM garage
                                    WHERE naam = 'Garage De Leeuw')
                  AND NOT EXISTS (SELECT * FROM interventie i2
                                  WHERE wagen.wagenid = i2.wagenid
                                  AND i1.mechanicien = i2.mechanicien))
```

5. Maak de oefeningen 1, 2 en 5 van reeks III opnieuw met behulp van subqueries i.p.v. joins.

1. SELECT wagenid, nummerplaat, (SELECT naam FROM eigenaar WHERE wagen.eigenaarid = eigenaar.eigenaarid) as eigenaar
FROM wagen
2. SELECT DISTINCT mechanicien FROM interventies
WHERE garageid = (SELECT garageid FROM garage
 WHERE naam = 'All Cars')
5. SELECT * FROM model
WHERE NOT EXIST (SELECT * FROM wagen
 WHERE modelid = model.modelid)

Reeks V: groepering, aggregatiefuncties en having

1. Wat is de gemiddelde aankoopprijs van een wagen (rond af op 1 cent)? (11960,83)

```
SELECT ROUND(AVG(aankoopprijs),2) FROM wagen
```

2. Hoeveel interventies gebeurden er in 2000? (22)

```
SELECT count(*) FROM interventie  
WHERE EXTRACT(Year FROM datum) = 2000
```

3. Hoeveel verschillende types motoren zijn er beschikbaar in de modellen van 2005? (28)

```
SELECT count(distinct type)  
FROM motor INNER JOIN model USING(modelid)  
WHERE jaar = 2005
```

4. Geef voor elke postcode het aantal wagens. (9 records)

```
SELECT postcode, count(*)  
FROM eigenaar INNER JOIN wagen USING(eigenaarid)  
GROUP BY postcode
```

5. Hoeveel verschillende wagens werden er in 2004 gemiddeld gezien in één garage? (11,5)

```
SELECT avg(count(distinct wagenid)) FROM interventie  
WHERE EXTRACT(YEAR FROM datum) = 2004  
GROUP BY garageid
```

6. Geef per garage en per jaar de totaalomzet van de verkoop van wagens? (33 records) *Zoek naar een interventie met als omschrijving 'Aflevering' om te welke garage welke wagen verkocht.*

```
SELECT garageid, EXTRACT(YEAR FROM ingebruikname) jaar,  
       sum(aankoopprijs)  
FROM wagen INNER JOIN interventie USING(wagenid)  
WHERE omschrijving = 'Aflevering'  
GROUP BY garageid, EXTRACT(YEAR FROM ingebruikname)  
ORDER BY garageid, jaar
```

7. Welke wagens zijn de afgelopen 2 jaar reeds meer dan twee keer langs geweest in de garage? (7 records)

```
SELECT wagenid  
FROM wagen INNER JOIN interventie USING (wagenid)  
WHERE interventie.datum >= ADD_MONTHS(CURRENT_DATE, -24)  
GROUP BY wagenid  
HAVING count(interventieid) > 2
```

8. Geef de eigenaar met het grootste aantal wagens. (De Meulenaere Ben)

```
SELECT naam, count(wagenid)  
FROM wagen NATURAL JOIN eigenaar  
GROUP BY eigenaarid, naam  
HAVING count(wagenid) = (SELECT max(count(wagenid))  
                          FROM wagen  
                          GROUP BY eigenaarid)
```

Reeks VI: UNION, INTERSECT en MINUS operaties

1. Geef een lijst van adressen in de databank (eigenaars + garages) geef in een apart veld aan of het gaat om een garage dan wel een eigenaar. (56 records)

```
SELECT naam, adres, postcode, woonplaats, 'eigenaar' as type
FROM eigenaar UNION
SELECT naam, adres, postcode, woonplaats, 'garage' as type
FROM garage
```

2. Geef een lijst van eigenaars die zowel een Audi als een Volkswagen bezitten. (4 records)

```
SELECT eigenaarid, naam
FROM eigenaar INNER JOIN wagen USING(eigenaarid)
            INNER JOIN model USING(modelid)
WHERE merk = 'Audi'
INTERSECT
SELECT eigenaarid, naam
FROM eigenaar INNER JOIN wagen USING(eigenaarid)
            INNER JOIN model USING(modelid)
WHERE merk = 'Volkswagen'
```

3. Geef een lijst van motortypes die in de huidige modellen wel voor Audi maar niet voor Volkswagen beschikbaar zijn. (15 records)

```
SELECT type FROM motor INNER JOIN model USING(modelid)
WHERE jaar=EXTRACT(YEAR FROM CURRENT_DATE) AND merk='Audi'
MINUS
SELECT type FROM motor INNER JOIN model USING(modelid)
WHERE jaar=EXTRACT(YEAR FROM CURRENT_DATE)
AND merk='Volkswagen'
```

4. Geef een lijst van eigenaars (eigenaarid + naam) die klant zijn bij zowel garage 'Nearly New Cars' als 'Garage De Leeuw'. (6 records)

```
SELECT eigenaarid, eigenaar.naam
FROM eigenaar INNER JOIN wagen USING(eigenaarid)
            INNER JOIN interventie USING(wagenid)
            INNER JOIN garage USING(garageid)
WHERE garage.naam='Garage De Leeuw'
INTERSECT
SELECT eigenaarid, eigenaar.naam
FROM eigenaar INNER JOIN wagen USING(eigenaarid)
            INNER JOIN interventie USING(wagenid)
            INNER JOIN garage USING(garageid)
WHERE garage.naam='Nearly New Cars'
```

Reeks VII: data dictionary

De tabellen die worden gebruikt in de oefeningen behoren tot het schema 'DATABANKEN'. De interface maakt gebruik van de user 'STUDENTDB'

1. Geef een lijst van alle tabelnamen van de gebruiker DATABANKEN. (6 records)

```
SELECT table_name FROM all_tables
WHERE owner='DATABANKEN'
```

2. Geef een lijst van tabellen waarop de gebruiker STUDENTDB select privileges heeft. (6 records)

```
SELECT table_name FROM user_tab_privs
WHERE privilege = 'SELECT' AND grantee='STUDENTDB'
```

3. Geef een lijst van indexen in het schema DATABANKEN. Vermeld het schema, de indexnaam, de naam van de tabel waarop de index is gedefiniëerd en het aantal kolommen van de index. (9 records)

```
SELECT owner, index_name, i.table_name, count(*)
FROM all_indexes i INNER JOIN all_ind_columns c
                        USING(index_name)
WHERE owner = 'DATABANKEN'
GROUP BY owner, index_name, i.table_name
```


Reeks VIII: Aanpassingen in de databank

1. Voeg jezelf toe aan de databank als eigenaar.

```
INSERT INTO eigenaar
  (eigenaarid, naam, adres, postcode, woonplaats)
VALUES (99, 'Jan Desmet', 'Noordstr. 14', '6345', 'Randomstad')
```

2. Nu je toch eigenaar bent, kan je jezelf een mooie wagen cadeau doen. Voeg ook de aflevering toe aan de databank.

```
INSERT INTO wagen (wagenid, chassisnr, nummerplaat,
  ingebruikname, modelid, motorid, eigenaarid, aankoopprijs)
VALUES (1234, 'ZEDIU672D2X3', 'ZDF-654', '21-NOV-2005', 48, 3,
  99, 28324)

# interventieid wordt automatisch ingevuld
INSERT INTO interventie (wagenid, garageid, datum, kmstand,
  mechanicien, type, omschrijving)
VALUES (1234, 4, '21-NOV-2005', 0, 'Pieter',
  'onderhoud', 'Aflevering')
```

3. Jammer genoeg heb je een maandags exemplaar gekregen dat regelmatig in panne valt. Voeg enkele interventies toe voor het nakijken en depanneren van je wagen.

```
INSERT INTO interventie (wagenid, garageid, datum, kmstand,
  mechanicien, type, omschrijving)
VALUES (1234, 4, '10-DEC-2005', 276, 'Mieke',
  'herstelling', 'Geblokkeerde versnellingsbak')
```

4. Je bent verhuisd. Pas je adres aan in de databank.

```
UPDATE eigenaar SET adres = 'Stationstraat 15', postcode =
  '7521', woonplaats = 'Mariakerke' WHERE eigenaarid = 99
```

5. Men is vergeten de BTW aan te rekenen op je wagen. Verhoog het bedrag met 21%.

```
UPDATE wagen SET aankoopprijs = 1.21 * aankoopprijs
  WHERE wagenid = 1234
```

6. Dit was helaas slechts een droom. Verwijder alle gegevens die je zelf hebt toegevoegd uit de databank.

```
DELETE FROM interventie WHERE wagenid = 1234
DELETE FROM wagen WHERE wagenid = 1234
DELETE FROM eigenaar WHERE eigenaarid = 99
```

Reeks IX: herhaling

1. Geef een lijst van Volkswagen eigenaars (naam, adres en modelnaam) verkocht tussen 21-03-2004 en 16-9-2005. (7 records)

```
SELECT naam, adres, postcode, woonplaats, modelnaam
FROM wagen INNER JOIN model USING(modelid)
      INNER JOIN eigenaar USING(eigenaarid)
WHERE ingebruikname BETWEEN TO_DATE('21-MAR-2004')
                        AND TO_DATE('16-SEP-2005')
      AND merk = 'Volkswagen'
```

2. Geef per model(naam) en per jaar het aantal verkochte exemplaren.

```
SELECT merk, modelnaam, jaar, count(*)
FROM Model LEFT JOIN wagen USING(modelid)
GROUP BY modelid, jaar, merk, modelnaam
ORDER BY jaar, merk, modelnaam
```

3. Geeft voor **elk** model uit 2001 het merk, de modelnaam, het jaar en het gemiddelde aantal onderhoudsbeurten per wagen. Rond het gemiddelde af op twee decimalen. (7 records)

```
SELECT merk, modelnaam, jaar, round(avg(aantal), 2)
FROM model LEFT JOIN (SELECT modelid, wagenid, count(*) aantal
                     FROM interventie INNER JOIN wagen
                                     USING(wagenid)
                     WHERE type='onderhoud'
                     GROUP BY modelid, wagenid) USING(modelid)
WHERE jaar = 2001
GROUP BY modelid, merk, modelnaam, jaar
```

4. Geef het gemiddeld aantal afgelegde kilometers tussen twee onderhoudsbeurten.

```
SELECT AVG(kmstand-(SELECT max(kmstand) FROM interventie i2
                    WHERE i2.wagenid = i1.wagenid
                    AND type = 'onderhoud'
                    AND i2.datum < i1.datum)) gemiddelde
FROM interventie i1 WHERE kmstand > 0 AND type='onderhoud'
```

5. Geef een lijst van wagens waarvan het jaar van het model niet overeenkomt met de datum van ingebruikname. (7 records)

```
SELECT * FROM wagen INNER JOIN model USING(modelid)
WHERE jaar <> EXTRACT(YEAR FROM ingebruikname)
```

6. Geef voor elke garage het meest verkochte merk. (9 records)

```
SELECT garageid, merk, count(wagenid)
FROM model INNER JOIN wagen USING(modelid)
      INNER JOIN interventie i USING(wagenid)
      INNER JOIN garage g ON i.garageid = g.garageid
WHERE omschrijving = 'Aflevering'
GROUP BY garageid, merk
HAVING count(wagenid)
      = (SELECT max(count(wagenid))
        FROM interventie INNER JOIN wagen USING(wagenid)
                        INNER JOIN model USING(modelid)
        WHERE garageid = g.garageid AND omschrijving='Aflevering'
        GROUP BY merk)
ORDER BY garageid, merk
```