

Oefeningen op eenvoudige Selects

Run het create en insert script van de onderneming database.
Voer de volgende wijziging door in de tabel MEDEWERKERS:

```
UPDATE medewerkers  
SET plaats='maastricht'  
WHERE sofi_nr='999555555';  
COMMIT;
```

1. Geef alle gegevens over alle projecten die binnen het bedrijf worden gerealiseerd.

	PROJ_NR	PROJ_NAAM	LOCATIE	AFD_NR
1	1	Orderverwerking	Oegstgeest	7
2	2	Salarisadministratie	Groningen	7
3	3	Magazijn	Eindhoven	7
4	10	Inventaris	Maastricht	3
5	20	Personeelszaken	Eindhoven	1
6	30	Debiteuren	Maastricht	3

```
SELECT *  
FROM projecten;
```

2. Druk voor alle projecten de projectnaam en het afdelingnummer van de ondersteunende afdeling af

	PROJ_NAAM	AFD_NR
1	Orderverwerking	7
2	Salarisadministratie	7
3	Magazijn	7
4	Inventaris	3
5	Personeelszaken	1
6	Debiteuren	3

```
SELECT proj_naam,afd_nr  
FROM projecten;
```

3.
a. Wijzig de voorgaande select zodat je de volgende resultatentabel krijgt:

	PROJECT	PROJ_NR	WORDTONDERSTEUNDOORAFDELING	AFD_NR
1	project	1	wordt ondersteund door afdeling	7
2	project	2	wordt ondersteund door afdeling	7
3	project	3	wordt ondersteund door afdeling	7
4	project	10	wordt ondersteund door afdeling	3
5	project	20	wordt ondersteund door afdeling	1
6	project	30	wordt ondersteund door afdeling	3

```
SELECT 'project ',proj_nr, 'wordt ondersteund door afdeling',
proj_afd_nr
FROM PROJECT;
```

b. Zorg er nu voor dat de hoofdingen voor de constante kolommen blanco blijven:

	PROJECT	PROJ_NR	WORDTONDERSTEUNDOORAFDELING	AFD_NR
1	project	1	wordt ondersteund door afdeling	7
2	project	2	wordt ondersteund door afdeling	7
3	project	3	wordt ondersteund door afdeling	7
4	project	10	wordt ondersteund door afdeling	3
5	project	20	wordt ondersteund door afdeling	1
6	project	30	wordt ondersteund door afdeling	3

```
SELECT 'project "'",proj_nr, 'wordt ondersteund door afdeling' " ",
afd_nr
FROM PROJECTEN;
```

Opm: om géén cijfer te krijgen in de alias, bij de eerste " " 1 spatie zetten en bij de tweede " " 2 spaties zetten.

c. Zorg ervoor dat je alles in 1 kolom afdrukt en geef de ganse kolom de hoofding "projecten met afdeling"

	projecten met afdeling
1	project 1 wordt ondersteund door afdeling 7
2	project 2 wordt ondersteund door afdeling 7
3	project 3 wordt ondersteund door afdeling 7
4	project 10 wordt ondersteund door afdeling 3
5	project 20 wordt ondersteund door afdeling 1
6	project 30 wordt ondersteund door afdeling 3

```
SELECT 'project '||proj_nr||' wordt ondersteund door afdeling ' ||
afd_nr "projecten met afdeling "
```

FROM PROJECTEN;

4.

Voer de volgende instructie uit. Wat geeft het resultaat weer?

```
SELECT SYSDATE- geb_datum  
FROM gezinsleden;
```

⇒ *Leeftijd gezinslid in dagen uitgedrukt*

5.

Verklaar telkens de fout:

```
SELECT *  
FROM opdracht;  
⇒ foutieve tabel
```

```
SELECT sofi_nr,proj_nr,uren;  
⇒ FROM ontbreekt
```

```
SELECT achternaam, salaris afd_nr  
FROM medewerkers;  
⇒ geen komma tussen 2 laatste  
attributen in SELECT
```

```
SELECT afd_nr, mgr_sofi_nr, datum_in_dienst  
FROM afdelingen  
⇒ datum_in_dienst is geen attribuut uit de tabel
```

6.

Er wordt een overzicht gevraagd van de woonplaatsen van de medewerkers van het bedrijf. Dit geeft onderstaande resultatentabel :

	PLAATS
1	Eindhoven
2	Maastricht
3	Oegstgeest
4	Maarssen
5	Groningen
6	maastricht

```
SELECT DISTINCT plaats
FROM medewerkers;
```

Bekijk de resultatentabel kritisch en schrijf een query die een beter resultaat geeft:

```
SELECT DISTINCT(UPPER(plaats))
FROM medewerkers;
```

➔ zal alles in de RT ook in hoofdletters omzetten

OF

```
SELECT DISTINCT(Initcap(plaats)) plaats
FROM medewerkers;
```

	PLAATS
1	Eindhoven
2	Maastricht
3	Oegstgeest
4	Maarssen
5	Groningen

7.

We willen weten in welke afdelingen medewerkers tewerkgesteld zijn en wat hun woonplaats is. Schrijf de query.

	AFD_NR	PLAATS
1	1	Eindhoven
2	7	Eindhoven
3	7	Oegstgeest
4	7	Groningen
5	3	Maastricht
6	3	Maarssen

```
SELECT DISTINCT afd_nr,INITCAP(plaats) plaats
FROM medewerkers;
```

8.

Dual is een tabel die standaard in de database aanwezig is. Ze bestaat uit 1 kolom en kan bv gebruikt worden om te testen:

```
SELECT sysdate
FROM dual;
```

```
SELECT TRUNC((50 -45)/3) resultaat
FROM dual;
```

RESULTAAT
1

Gebruik nu de tabel DUAL om met de afzonderlijke woorden "SQL", "Data Retrieval", "hoofdstuk 3-4" in één instructie de volgende resultaatentabel te bekomen:

Beste Klas
SQL Data retrieval hoofdstuk 3-4

```
SELECT 'SQL'||' '||'Data retrieval'||' '||'hoofdstuk 3-4' "Beste Klas"
FROM dual;
```

OF

```
SELECT 'SQL Data retrieval hoofdstuk 3-4' "Beste Klas"
FROM dual;
```

9.

Welke Personen horen tot het gezin van medewerker 999111111?
Je moet exact de volgende resultaatentabel bekomen.

MEDEWERKER	NAAM GEZINSLID	RELATIE	GESLACHT
999111111	Diana	DOCHTER	V
999111111	Jos	ZOON	M
999111111	Mary	PARTNER	V

```
SELECT sofi_nr medewerker, naam "NAAM GEZINSLID", relatie,
geslacht
FROM gezinsleden
WHERE sofi_nr='999111111'
ORDER BY 2;
```

10.

Geef alle informatie over de afdeling administratie?

AFD_NR	AFD_NAAM	MGR_SOFI_NR	MGR_START_DATUM
3	Administratie	999555555	01/01/11

```
SELECT *
FROM afdelingen
WHERE UPPER(afd_naam) = 'ADMINISTRATIE';
```

11.

We willen een overzicht van alle medewerkers die uit Maastricht afkomstig zijn en geven daartoe de volgende instructie:

```
SELECT sofi_nr, achternaam, plaats
FROM medewerkers
WHERE plaats='Maastricht';
```

Dit geeft de volgende resultatentabel:

	SOFI_NR	ACHTERNAAM	PLAATS
1	999222222	Amelsvoort	Maastricht

Waarom zit medewerker 999555555 niet in de resultatentabel?

Hoofdlettergevoelig

Pas de query op een efficiënte manier aan.

```
SELECT sofi_nr, achternaam, plaats
FROM medewerkers
WHERE INITCAP(plaats)='Maastricht';
```

12.

Welke medewerkers werkten aan project 10 tussen de 20 en 35 uren (beide inclusief)?

Geef het sofi_nr, het projectnr en het aantal gepresteerde uren weer.

	SOFI_NR	PROJ_NR	UREN
1	999222222	10	34,5

```
SELECT sofi_nr, proj_nr, uren
FROM opdrachten
WHERE uren BETWEEN 20 AND 35
AND proj_nr=10;
```

13.

Aan welke projecten werkte medewerker 999222222 minder dan 10 uren? Geef projnr en aantal uren.

	PROJ_NR	UREN
1	30	5,1

```
SELECT proj_nr, uren
FROM opdrachten
Where uren < 10 AND sofi_nr='999222222';
```

14.

Welke medewerkers komen uit de Provincie Groningen (GR) of Noord Brabant (NB)? Los op 2 manieren op!

	SOFI_NR	ACHTERNAAM	PROVINCIE
1	999666666	Bordoloi	NB
2	999333333	Joosten	GR
3	999888888	Pregers	NB

```
SELECT sofi_nr, achternaam, provincie
FROM medewerkers
WHERE UPPER(provincie) IN ('NB','GR');
```

```
SELECT sofi_nr, achternaam, provincie
FROM medewerkers
WHERE UPPER(provincie) = 'NB' OR UPPER(provincie) = 'GR';
```

15.

Zijn er medewerkers met voornaam Suzan, Martina, Henk of Douglas en op welke afdeling werken ze? Sorteer op afd_nr (in dalende volgorde) en daarbinnen alfabetisch op voornaam.

	AFD_NR	VOORNAAM
1	7	Douglas
2	3	Henk
3	3	Martina
4	3	Suzan

```
SELECT afd_nr, voornaam
FROM medewerkers
WHERE UPPER(voornaam) IN
('SUZAN','MARTINA','HENK','DOUGLAS')
ORDER BY afd_nr DESC, voornaam;
```

16.

We willen in onze resultatentabel:

Naam, afdelingnr en salaris van medewerkers uit afdeling 7 die minder dan 40000 verdienen
+ de naam, het afdelingnr en het salaris van medewerker 999666666

	ACHTERNAAM	SALARIS	AFD_NR
1	Bordoloi	55000	1
2	Bock	30000	7
3	Joosten	38000	7
4	Pregers	25000	7

```
SELECT achternaam, salaris, afd_nr
FROM medewerkers
WHERE afd_nr=7 AND salaris<40000 OR sofi_nr='999666666';
```

Waarom zijn er voor je oplossing geen haakjes nodig?
Omdat AND sowieso voorrang heeft op OR

17.

Welke medewerkers wonen niet in Maarssen en ook niet in Eindhoven?

	ACHTERNAAM	AFD_NR
1	Jochems	3
2	Zuiderweg	7
3	Amelsvoort	3
4	Bock	7
5	Joosten	7

```
SELECT achternaam, afd_nr
FROM medewerkers
WHERE UPPER(plaats) NOT IN ('MAARSSSEN','EINDHOVEN');
```

of

```
WHERE UPPER(plaats) != 'MAARSSSEN' AND UPPER(plaats) !=
'EINDHOVEN';
```

of

```
WHERE NOT UPPER(plaats) = 'MAARSSSEN' AND NOT UPPER(plaats)
= 'EINDHOVEN';
```

18.

Geef een overzicht van de inhoud van de tabel OPDRACHTEN.

a.

Sorteer op UREN in stijgende volgorde.

Zorg dat rijen met een NULL waarde voor UREN als eerste getoond worden.

	SOFI_NR	PROJ_NR	UREN
1	999666666	20	(null)
2	999444444	1	(null)
3	999222222	30	5,1
4	999111111	2	8,5
5	999444444	10	10,1
6	999887777	10	10,2
7	999444444	3	10,5
8	999444444	20	11,8
9	999444444	2	12,2
10	999555555	20	14,8
11	999555555	30	19,2
12	999888888	1	21
13	999888888	2	22
14	999887777	30	30,8
15	999111111	1	31,4
16	999222222	10	34,5
17	999333333	3	42,1

```
SELECT *
FROM opdrachten
ORDER BY uren ASC NULLS FIRST;
```

b. Sorteer op OPD_UREN in dalende volgorde.
Zorg dat rijen met een NULL waarde voor OPD_UREN als laatste getoond worden.

	SOFI_NR	PROJ_NR	UREN
1	999333333	3	42,1
2	999222222	10	34,5
3	999111111	1	31,4
4	999887777	30	30,8
5	999888888	2	22
6	999888888	1	21
7	999555555	30	19,2
8	999555555	20	14,8
9	999444444	2	12,2
10	999444444	20	11,8
11	999444444	3	10,5
12	999887777	10	10,2
13	999444444	10	10,1
14	999111111	2	8,5
15	999222222	30	5,1
16	999444444	1	(null)
17	999666666	20	(null)

```
SELECT *  
FROM opdrachten  
ORDER BY uren DESC NULLS LAST;
```