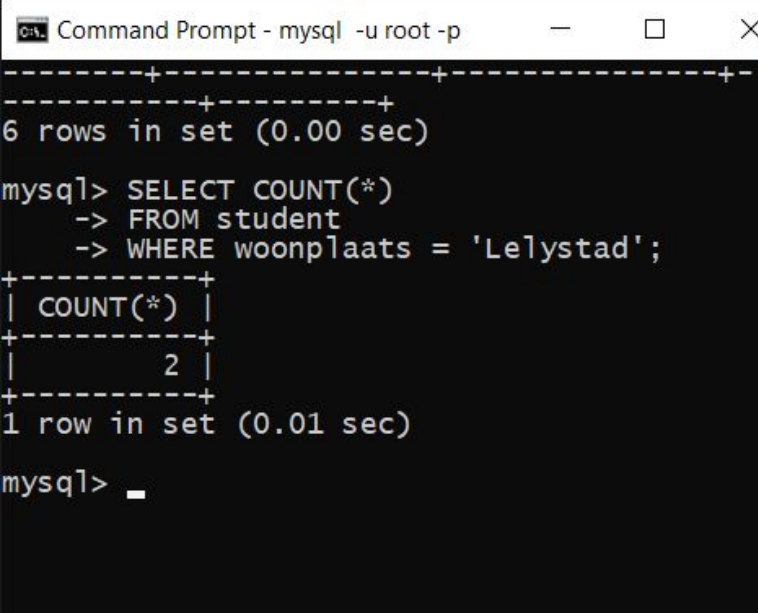


ANTWOORDEN HOOFDSTUK 6

Opdracht 6.1

```
SELECT COUNT(*)  
-> FROM student  
-> WHERE woonplaats = 'Lelystad';
```

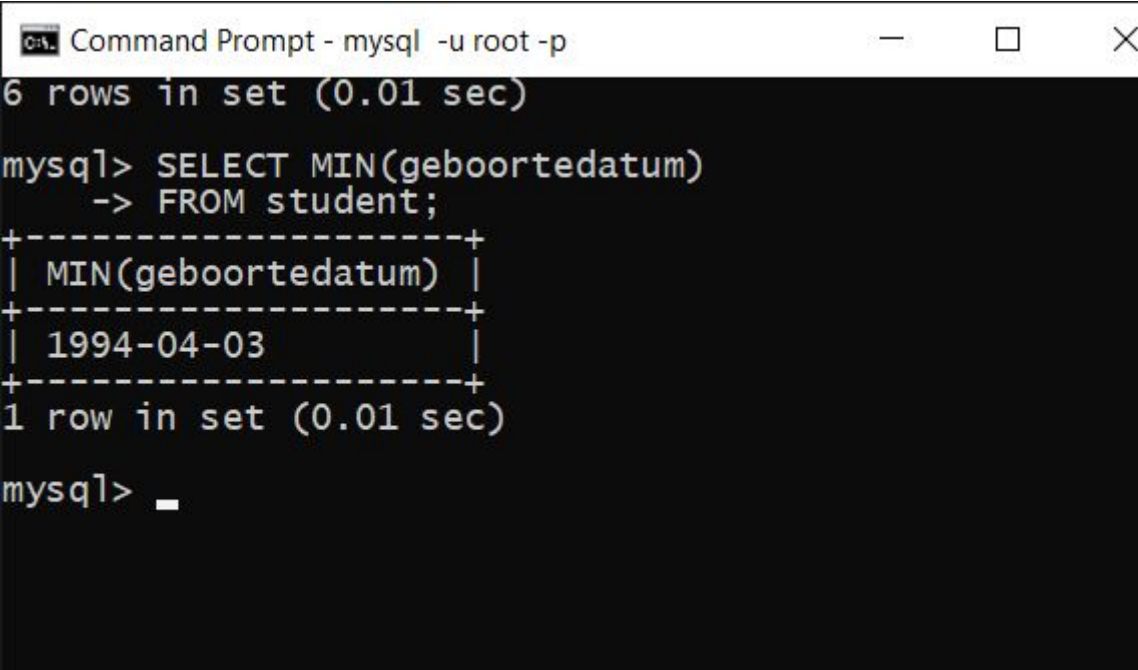


The screenshot shows a Windows Command Prompt window titled "Command Prompt - mysql -u root -p". The prompt is at the mysql> level. The user has entered the query: `SELECT COUNT(*) FROM student WHERE woonplaats = 'Lelystad';`. The output shows "6 rows in set (0.00 sec)" followed by the query execution. The result is displayed in a table format with a single row containing the value 2. The prompt then shows "1 row in set (0.01 sec)" and the mysql> prompt with a cursor.

```
-----+-----  
-----+-----  
6 rows in set (0.00 sec)  
  
mysql> SELECT COUNT(*)  
      -> FROM student  
      -> WHERE woonplaats = 'Lelystad';  
+-----+  
| COUNT(*) |  
+-----+  
|      2   |  
+-----+  
1 row in set (0.01 sec)  
  
mysql> _
```

Opdracht 6.2

```
SELECT MIN(geboortedatum)  
-> FROM student;
```

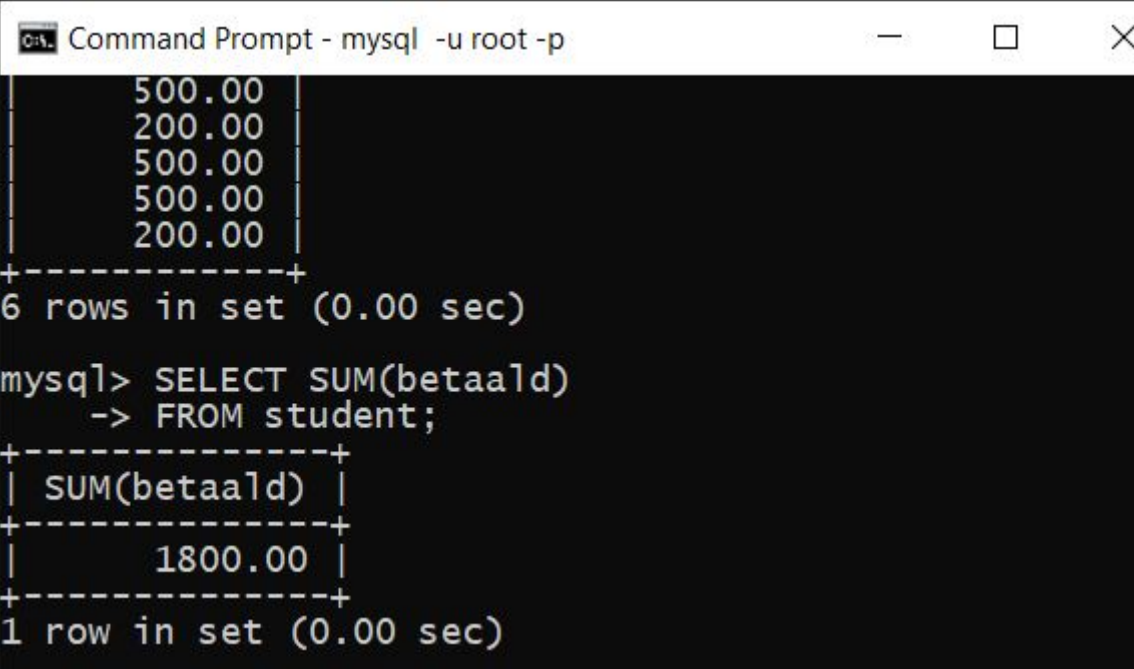


The screenshot shows a Windows Command Prompt window titled "Command Prompt - mysql -u root -p". The prompt is at the mysql> level. The user has entered the query: `SELECT MIN(geboortedatum) FROM student;`. The output shows "6 rows in set (0.01 sec)" followed by the query execution. The result is displayed in a table format with a single row containing the date 1994-04-03. The prompt then shows "1 row in set (0.01 sec)" and the mysql> prompt with a cursor.

```
-----+-----  
-----+-----  
6 rows in set (0.01 sec)  
  
mysql> SELECT MIN(geboortedatum)  
      -> FROM student;  
+-----+  
| MIN(geboortedatum) |  
+-----+  
| 1994-04-03         |  
+-----+  
1 row in set (0.01 sec)  
  
mysql> _
```

Opdracht 6.3

```
SELECT SUM(betaald)
-> FROM student;
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt - mysql -u root -p". The window displays the output of a MySQL query. The first part shows a table with 6 rows of data, each with a single column containing a numerical value. The second part shows the execution of the query `SELECT SUM(betaald) FROM student;`, which returns a single row with the sum of the values in the `betaald` column.

500.00
200.00
500.00
500.00
200.00
500.00

6 rows in set (0.00 sec)

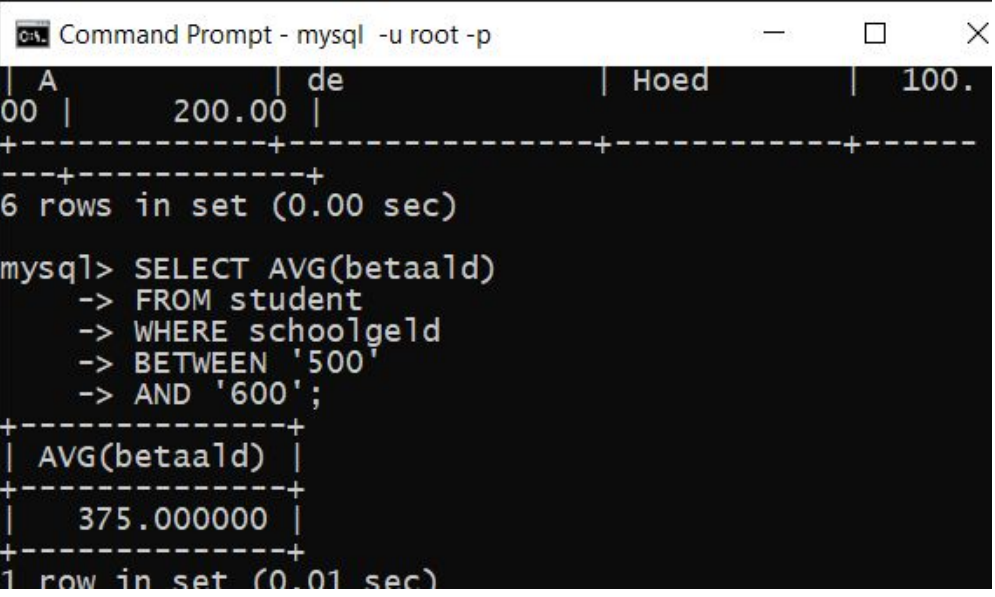
```
mysql> SELECT SUM(betaald)
-> FROM student;
```

SUM(betaald)
1800.00

1 row in set (0.00 sec)

Opdracht 6.4

```
SELECT AVG(betaald)
-> FROM student
-> WHERE schoolgeld
-> BETWEEN '500'
-> AND '600';
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt - mysql -u root -p". The window displays the output of a MySQL query. The first part shows a table with 6 rows of data, each with three columns: a letter, a numerical value, and a text string. The second part shows the execution of the query `SELECT AVG(betaald) FROM student WHERE schoolgeld BETWEEN '500' AND '600';`, which returns a single row with the average of the values in the `betaald` column for the specified range of `schoolgeld`.

A	de	Hoed
00	200.00	100.

6 rows in set (0.00 sec)

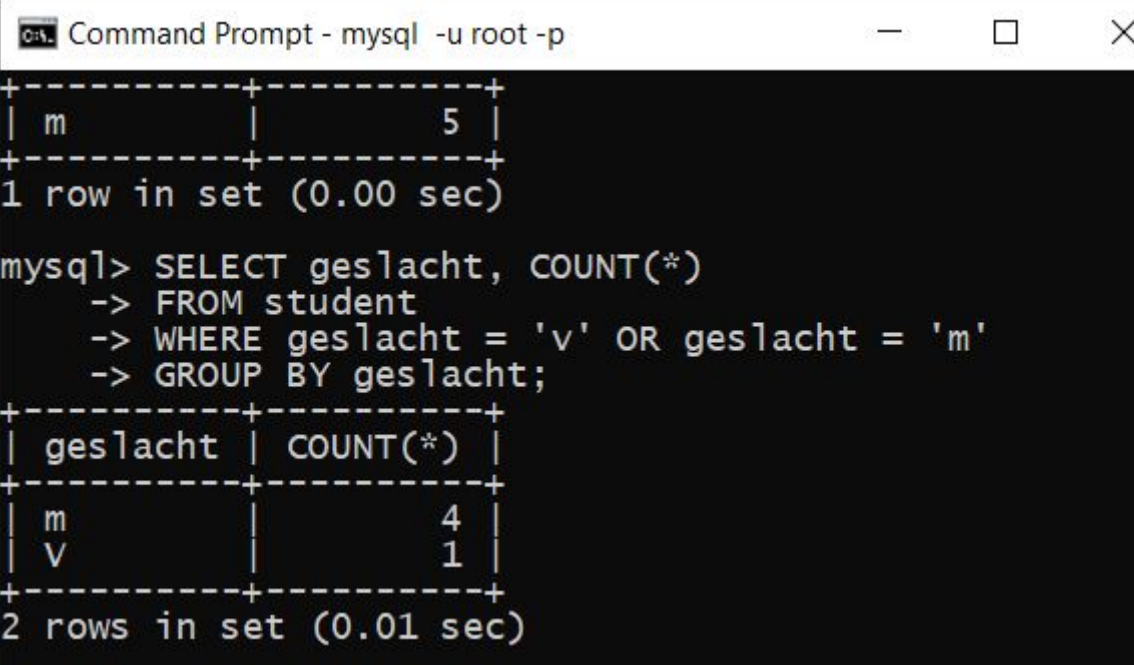
```
mysql> SELECT AVG(betaald)
-> FROM student
-> WHERE schoolgeld
-> BETWEEN '500'
-> AND '600';
```

AVG(betaald)
375.000000

1 row in set (0.01 sec)

Opdracht 6.5

```
SELECT geslacht, COUNT(*)  
-> FROM student  
-> WHERE geslacht = 'v' OR geslacht = 'm'  
-> GROUP BY geslacht;
```

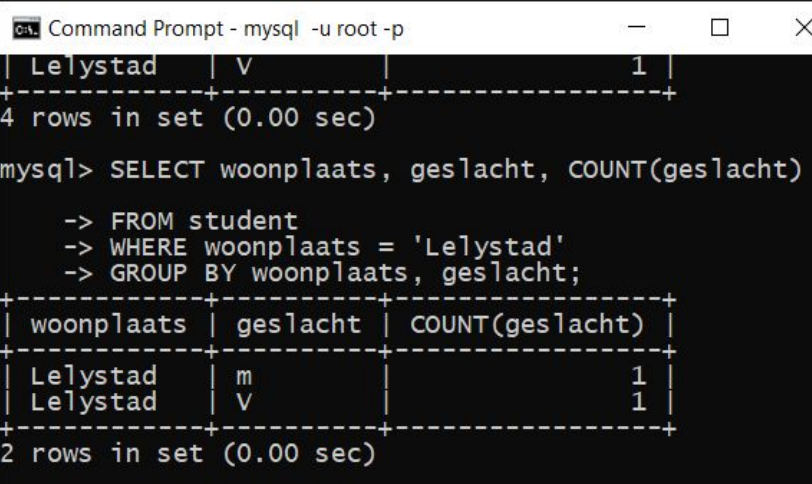


The screenshot shows a Windows Command Prompt window titled "Command Prompt - mysql -u root -p". The prompt shows the execution of a SQL query. The output is a table with two columns: 'geslacht' and 'COUNT(*)'. The first row shows 'm' with a count of 5. The second row shows 'v' with a count of 1. The prompt then shows the same query being executed again, but this time the output shows 'm' with a count of 4 and 'v' with a count of 1. This suggests a change in the data between the two executions.

```
mysql> SELECT geslacht, COUNT(*)  
-> FROM student  
-> WHERE geslacht = 'v' OR geslacht = 'm'  
-> GROUP BY geslacht;  
+-----+-----+  
| m      |      5 |  
+-----+-----+  
1 row in set (0.00 sec)  
  
mysql> SELECT geslacht, COUNT(*)  
-> FROM student  
-> WHERE geslacht = 'v' OR geslacht = 'm'  
-> GROUP BY geslacht;  
+-----+-----+  
| geslacht | COUNT(*) |  
+-----+-----+  
| m      |      4 |  
| v      |      1 |  
+-----+-----+  
2 rows in set (0.01 sec)
```

Opdracht 6.6

```
SELECT woonplaats, geslacht, COUNT(geslacht)  
-> FROM student  
-> WHERE woonplaats = 'Lelystad'  
-> GROUP BY woonplaats, geslacht;
```

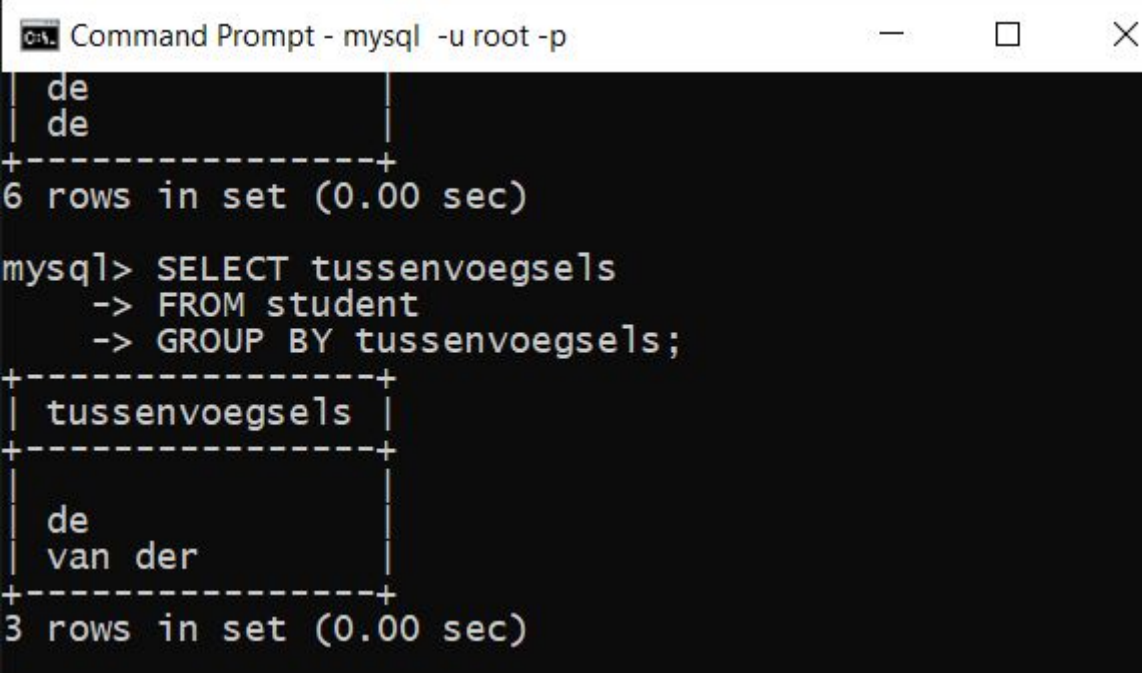


The screenshot shows a Windows Command Prompt window titled "Command Prompt - mysql -u root -p". The prompt shows the execution of a SQL query. The output is a table with three columns: 'woonplaats', 'geslacht', and 'COUNT(geslacht)'. The first row shows 'Lelystad' and 'v' with a count of 1. The second row shows 'Lelystad' and 'm' with a count of 1. The prompt then shows the same query being executed again, but this time the output shows 'Lelystad' and 'm' with a count of 1 and 'Lelystad' and 'v' with a count of 1. This suggests a change in the data between the two executions.

```
mysql> SELECT woonplaats, geslacht, COUNT(geslacht)  
-> FROM student  
-> WHERE woonplaats = 'Lelystad'  
-> GROUP BY woonplaats, geslacht;  
+-----+-----+-----+  
| Lelystad | v      |      1 |  
+-----+-----+-----+  
4 rows in set (0.00 sec)  
  
mysql> SELECT woonplaats, geslacht, COUNT(geslacht)  
-> FROM student  
-> WHERE woonplaats = 'Lelystad'  
-> GROUP BY woonplaats, geslacht;  
+-----+-----+-----+  
| woonplaats | geslacht | COUNT(geslacht) |  
+-----+-----+-----+  
| Lelystad   | m      |      1 |  
| Lelystad   | v      |      1 |  
+-----+-----+-----+  
2 rows in set (0.00 sec)
```

Opdracht 6.7

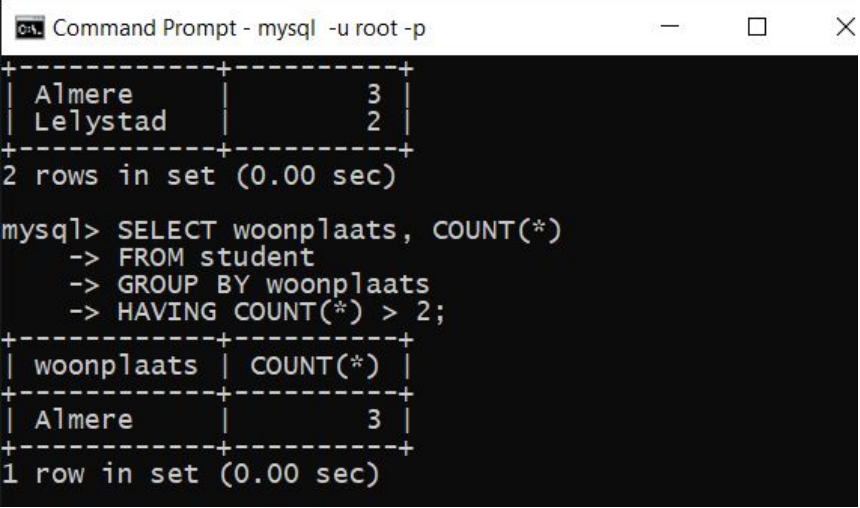
```
SELECT tussenvoegsels  
-> FROM student  
-> GROUP BY tussenvoegsels;
```



```
Command Prompt - mysql -u root -p  
de  
de  
+-----+  
6 rows in set (0.00 sec)  
  
mysql> SELECT tussenvoegsels  
-> FROM student  
-> GROUP BY tussenvoegsels;  
+-----+  
| tussenvoegsels |  
+-----+  
| de  
| van der  
+-----+  
3 rows in set (0.00 sec)
```

Opdracht 6.8

```
SELECT woonplaats, COUNT(*)  
-> FROM student  
-> GROUP BY woonplaats  
-> HAVING COUNT(*) > 2;
```



```
Command Prompt - mysql -u root -p  
+-----+-----+  
| Almere | 3 |  
| Lelystad | 2 |  
+-----+-----+  
2 rows in set (0.00 sec)  
  
mysql> SELECT woonplaats, COUNT(*)  
-> FROM student  
-> GROUP BY woonplaats  
-> HAVING COUNT(*) > 2;  
+-----+-----+  
| woonplaats | COUNT(*) |  
+-----+-----+  
| Almere | 3 |  
+-----+-----+  
1 row in set (0.00 sec)
```

Opdracht 6.9

```
SELECT woonplaats, SUM(schoolgeld - betaald)
-> FROM student
-> GROUP BY woonplaats;
```

```
Command Prompt - mysql -u root -p
+-----+-----+
| 600.00 |
| 0.00   |
+-----+-----+
3 rows in set (0.00 sec)

mysql> SELECT woonplaats, SUM(schoolgeld - betaald)
-> FROM student
-> GROUP BY woonplaats;
+-----+-----+
| woonplaats | SUM(schoolgeld - betaald) |
+-----+-----+
| Almere     | 0.00                      |
| Lelystad   | 600.00                    |
| Zeewolde   | 0.00                      |
+-----+-----+
3 rows in set (0.00 sec)
```

Opdracht 6.10

```
SELECT woonplaats, SUM(schoolgeld - betaald)
-> FROM student
-> WHERE woonplaats = 'Lelystad'
-> GROUP BY woonplaats
-> HAVING COUNT(*) > 1.00;
```

```
Command Prompt - mysql -u root -p
d)
-> FROM student
-> WHERE woonplaats = 'Lelystad'
-> GROUP BY woonplaats
-> HAVING COUNT(*) > 1.00;
+-----+-----+
| woonplaats | SUM(schoolgeld - betaald) |
+-----+-----+
| Lelystad   | 600.00                    |
+-----+-----+
1 row in set (0.00 sec)

mysql> _
```

