

# Функции агрегирования

Функция	Описание
AVG([ ALL DISTINCT ] <выражение>)	Вычисляет среднее арифметическое для заданного множества значений
COUNT( [ALL DISTINCT] <выражение>)	Вычисляет количество записей в таблице
MAX(<выражение>)	Вычисляет максимальное значение в заданном множестве
MIN(<выражение>)	Вычисляет минимальное значение в заданном множестве
SUM([ ALL DISTINCT ] <выражение>)	Вычисляет сумму указанных выражений в заданном множестве

# Особенности использования функций агрегирования

- В качестве выражения в агрегатных функциях может использоваться любая константа, функция, комбинация из названий столбцов, констант и функций, соединенных арифметическими или другими допустимыми в контексте данной СУБД операциями.
- Обработка значений NULL в агрегатных функциях по разному реализована в различных СУБД. Какие-то СУБД обрабатывают значение как 0, другие игнорируют его.
- По умолчанию в агрегатных функциях используется опция ALL.

# Простейший синтаксис для использования функций агрегирования

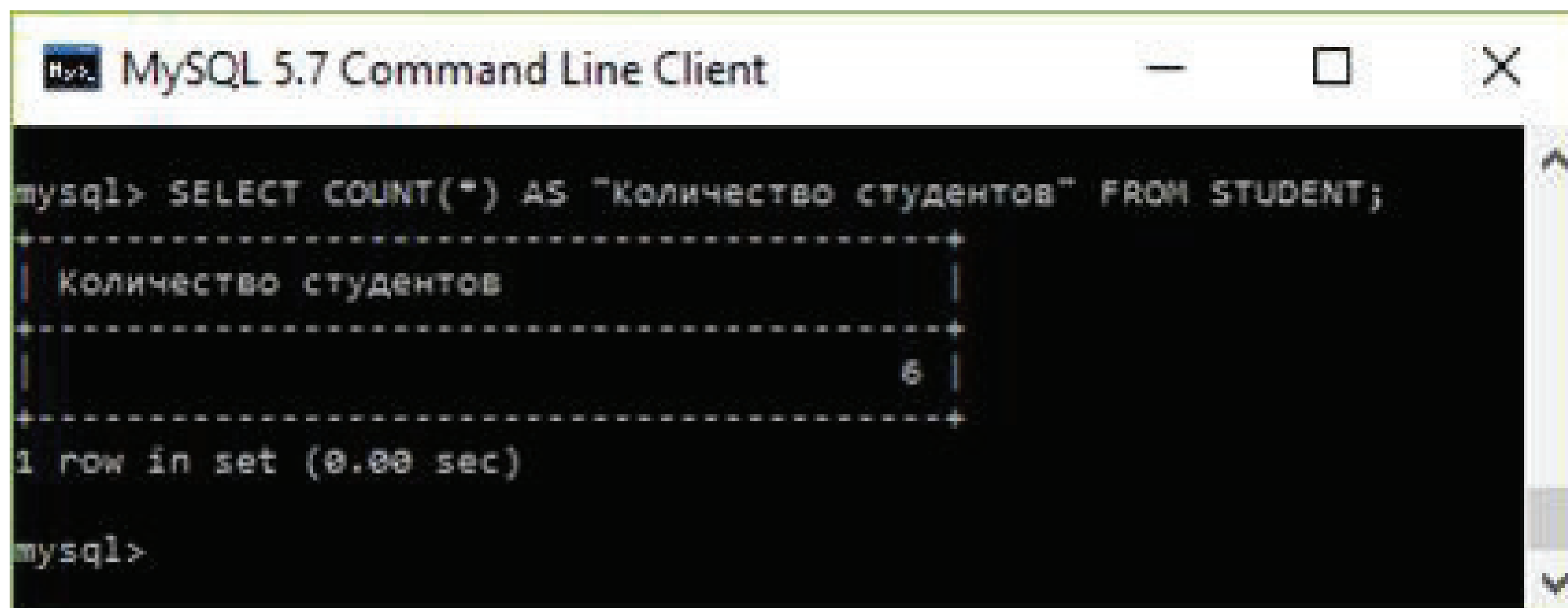
```
SELECT    aggr_function([ALL|DISTINCT] <выражение>)  
          { , aggr_function([ALL|DISTINCT] <выражение>) } *  
FROM <имя таблицы>
```

В большинстве СУБД при использовании хотя бы одной агрегатной функции недопустимо использование после SELECT выражений, отличных от агрегатных.

# Пример: простейшее агрегирование

Определение общего количества студентов.

```
SELECT COUNT(*) AS "Количество студентов" FROM STUDENT;
```



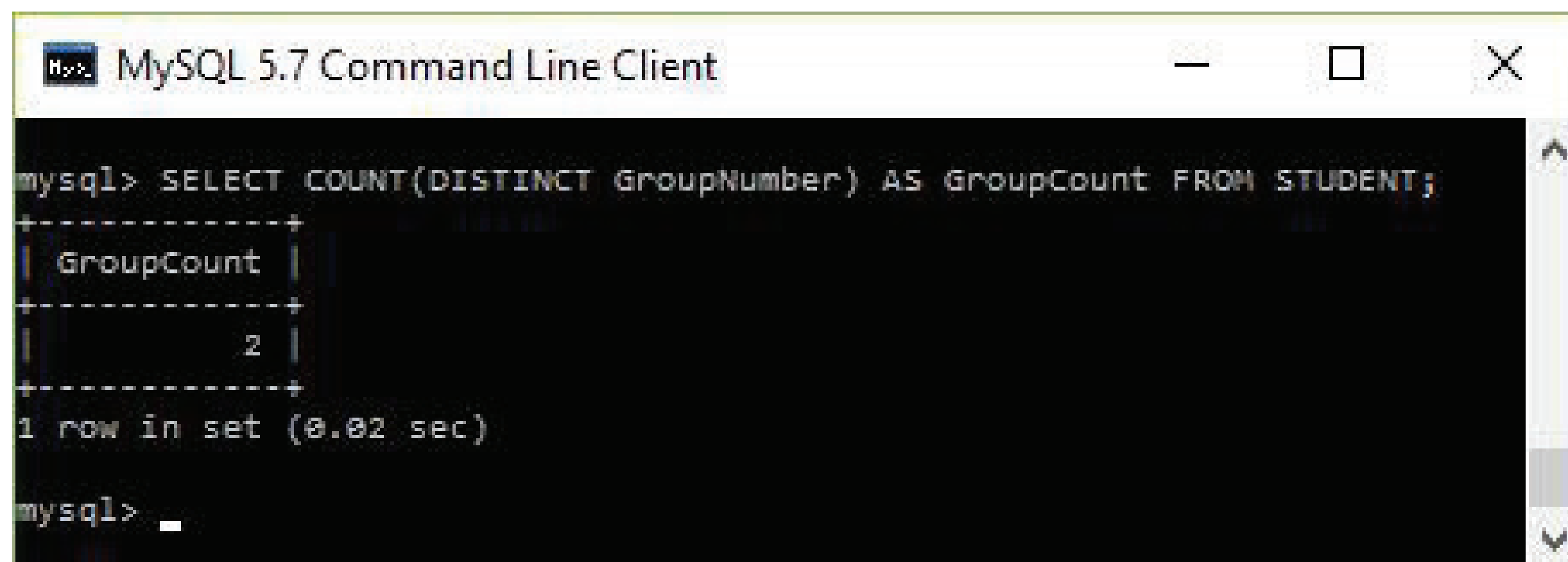
```
mysql> SELECT COUNT(*) AS "Количество студентов" FROM STUDENT;
+-----+
| Количество студентов |
+-----+
|                     6 |
+-----+
1 row in set (0.00 sec)

mysql>
```

# Пример: агрегирование с использованием опции факторизации

Количество групп, в которых учатся студенты.

```
SELECT COUNT(DISTINCT GroupNumber) AS GroupCount FROM STUDENT;
```



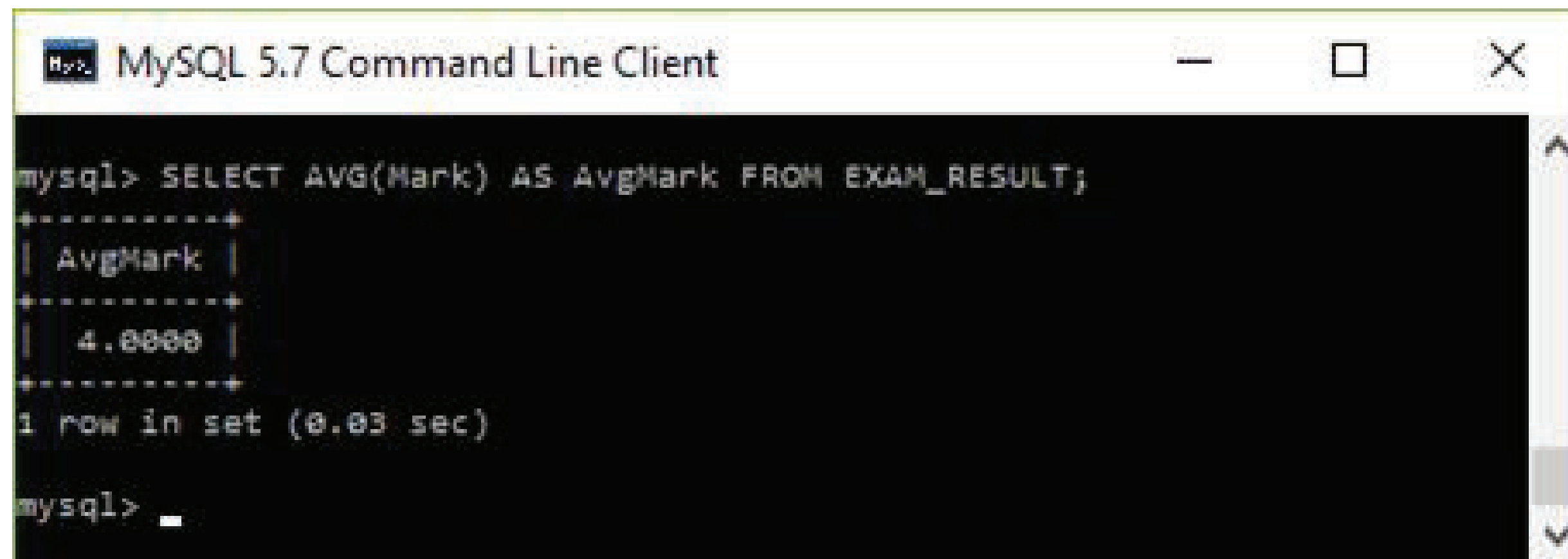
```
mysql> SELECT COUNT(DISTINCT GroupNumber) AS GroupCount FROM STUDENT;
+-----+
| GroupCount |
+-----+
|          2 |
+-----+
1 row in set (0.02 sec)

mysql> _
```

# Пример: простейшее агрегирование

Средний балл по итогам результатов экзаменов.

```
SELECT AVG(Mark) AS AvgMark FROM EXAM_RESULT;
```



```
MySQL 5.7 Command Line Client

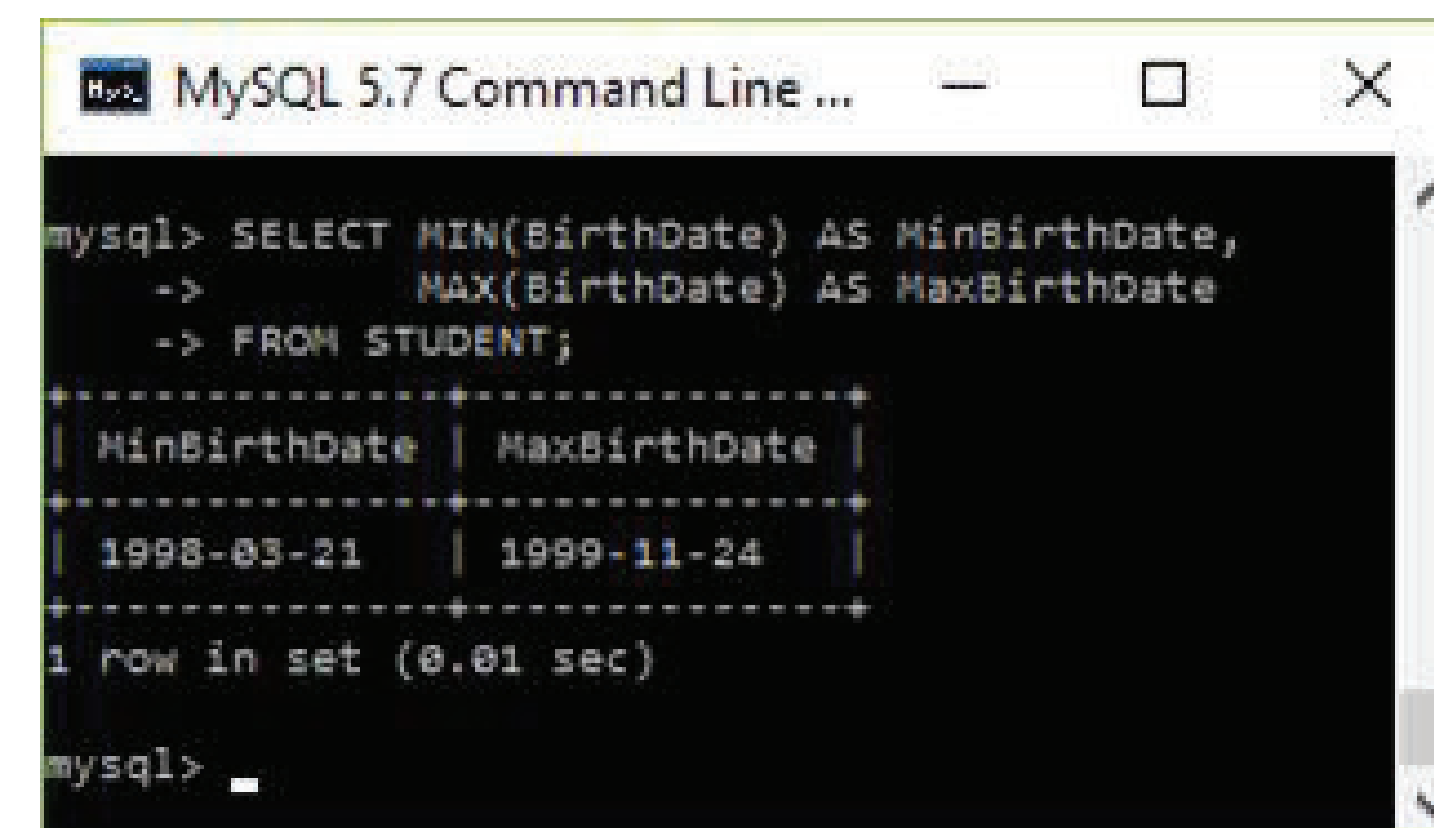
mysql> SELECT AVG(Mark) AS AvgMark FROM EXAM_RESULT;
+-----+
| AvgMark |
+-----+
| 4.0000 |
+-----+
1 row in set (0.03 sec)

mysql>
```

# Пример: агрегирование двух значений

Даты рождения самого младшего и самого старшего студентов.

```
SELECT MIN(BirthDate) AS MinBirthDate,  
       MAX(BirthDate) AS MaxBirthDate  
FROM STUDENT;
```



```
MySQL 5.7 Command Line ...  
mysql> SELECT MIN(BirthDate) AS MinBirthDate,  
->          MAX(BirthDate) AS MaxBirthDate  
-> FROM STUDENT;  
+-----+-----+  
| MinBirthDate | MaxBirthDate |  
+-----+-----+  
| 1998-03-21   | 1999-11-24   |  
+-----+-----+  
1 row in set (0.01 sec)  
  
mysql>
```

# Упрощенный синтаксис для использования функций агрегирования с группировкой

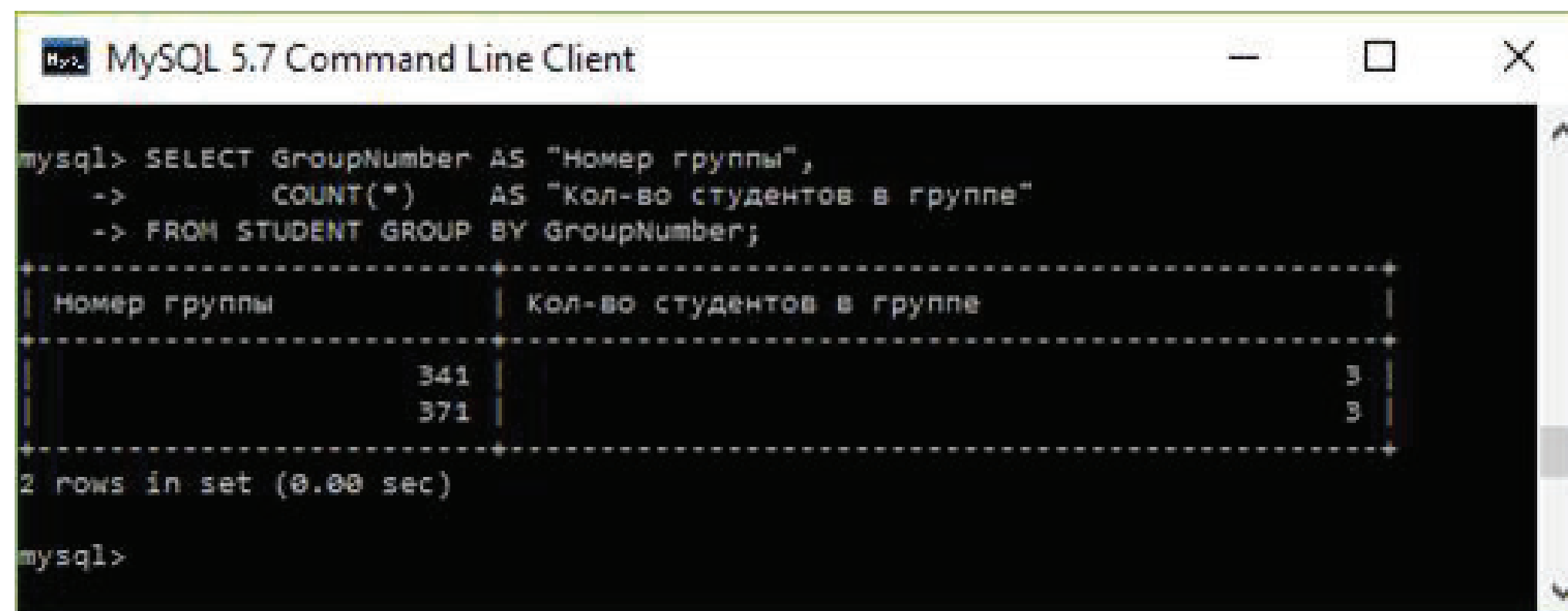
```
SELECT <список из агрегатных функций и выражений для группировки>  
FROM ...  
GROUP BY <выражения для группировки>
```



# Пример: агрегирование с группировкой по одному полю

Количество студентов в каждой группе.

```
SELECT GroupNumber AS "Номер группы",  
       COUNT(*)     AS "Кол-во студентов в группе"  
FROM STUDENT GROUP BY GroupNumber;
```



```
mysql> SELECT GroupNumber AS "Номер группы",  
->     COUNT(*)     AS "Кол-во студентов в группе"  
-> FROM STUDENT GROUP BY GroupNumber;
```

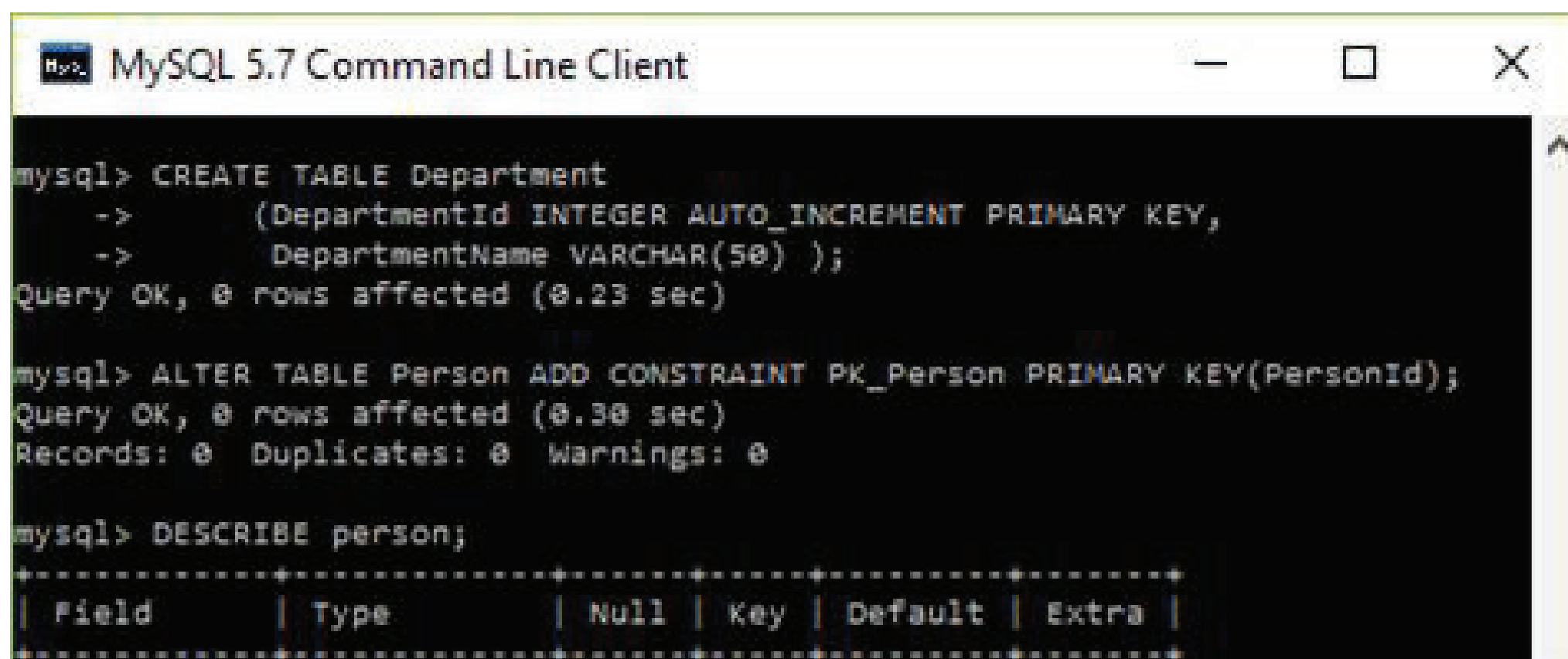
Номер группы	Кол-во студентов в группе
341	3
371	3

```
2 rows in set (0.00 sec)  
  
mysql>
```

# Пример: агрегирование с группировкой по двум полям

Сколько экзаменов назначено на каждую дату и аудиторию?

```
SELECT ClassRoom, ExamDate, COUNT(*) AS ExamCount  
FROM EXAM_SHEET  
GROUP BY ClassRoom, ExamDate;
```

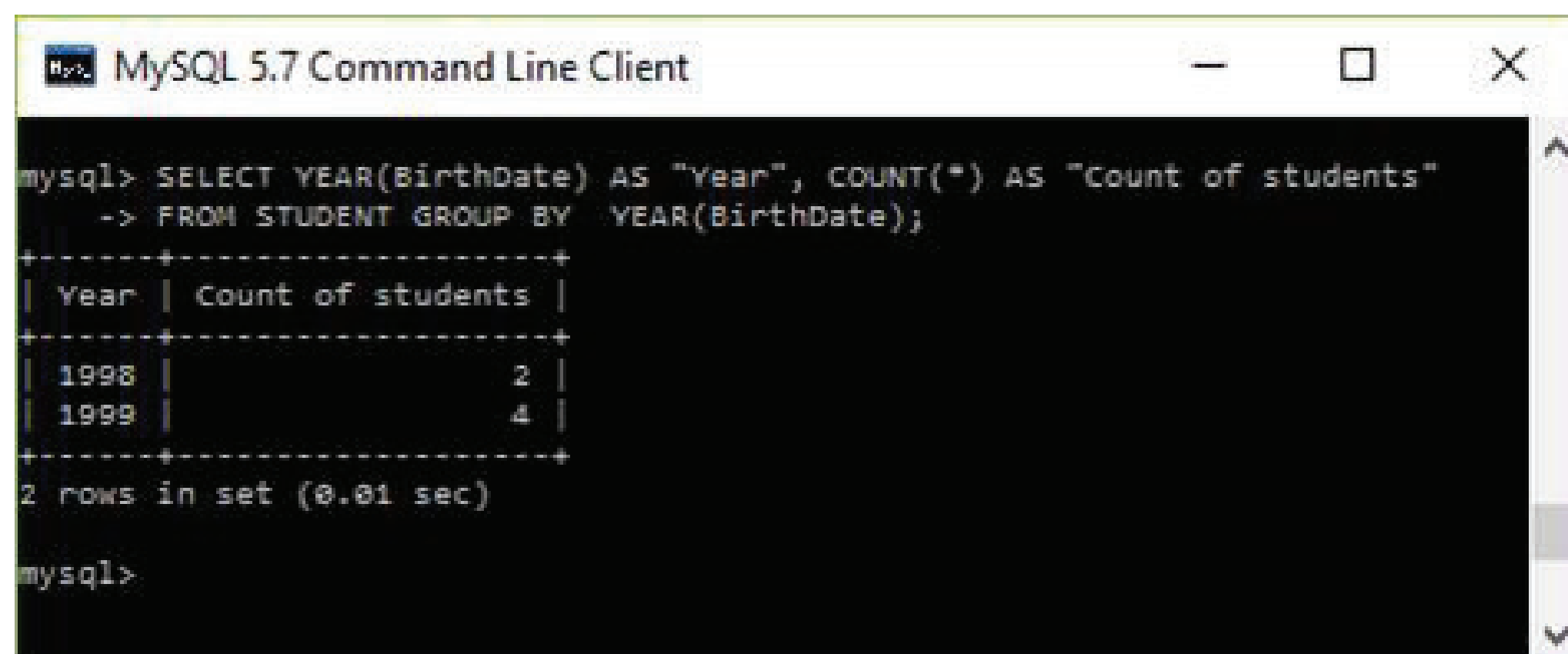


```
MySQL 5.7 Command Line Client  
mysql> CREATE TABLE Department  
-> (DepartmentId INTEGER AUTO_INCREMENT PRIMARY KEY,  
-> DepartmentName VARCHAR(50) );  
Query OK, 0 rows affected (0.23 sec)  
  
mysql> ALTER TABLE Person ADD CONSTRAINT PK_Person PRIMARY KEY(PersonId);  
Query OK, 0 rows affected (0.30 sec)  
Records: 0 Duplicates: 0 Warnings: 0  
  
mysql> DESCRIBE person;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+
```

# Пример: агрегирование с группировкой по значению функции

Как даты рождения студентов распределены по годам?

```
SELECT YEAR(BirthDate) AS "Year", COUNT(*) AS "Count of students"  
FROM STUDENT GROUP BY YEAR(BirthDate);
```



```
mysql> SELECT YEAR(BirthDate) AS "Year", COUNT(*) AS "Count of students"  
-> FROM STUDENT GROUP BY YEAR(BirthDate);  
+-----+-----+  
| Year | Count of students |  
+-----+-----+  
| 1998 | 2 |  
| 1999 | 4 |  
+-----+-----+  
2 rows in set (0.01 sec)  
  
mysql>
```

# Упрощенный синтаксис для агрегирования с условием для группировки

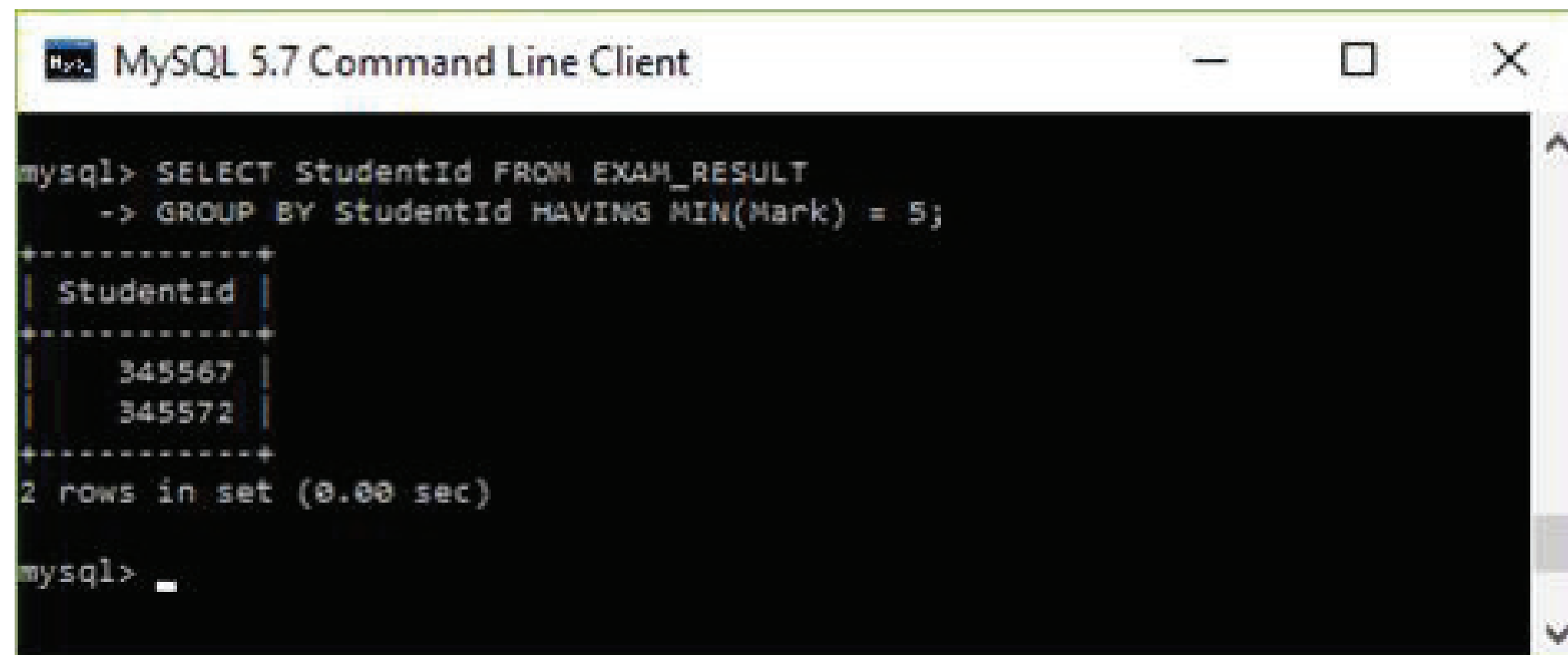
```
SELECT <список из агрегатных функций и выражений для группировки>  
FROM ...  
GROUP BY <выражения для группировки>  
HAVING <условие для группировки>
```

<условие для группировки> — это логическое выражение, построенное на основе агрегатных функций и <выражений для группировки>.

# Пример: агрегирование с условием для группировки

Номера зачетов студентов, получавших на экзаменах только 5.

```
SELECT StudentId FROM EXAM_RESULT  
GROUP BY StudentId HAVING MIN(Mark) = 5;
```



```
mysql> SELECT StudentId FROM EXAM_RESULT  
-> GROUP BY StudentId HAVING MIN(Mark) = 5;  
+-----+  
| StudentId |  
+-----+  
| 345567 |  
| 345572 |  
+-----+  
2 rows in set (0.00 sec)  
  
mysql> _
```