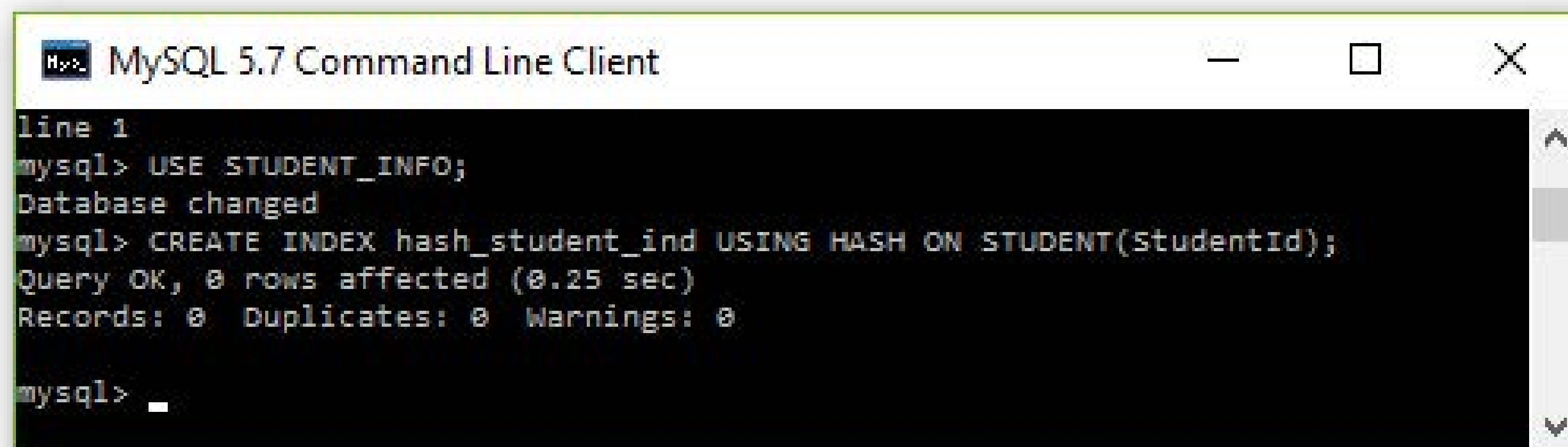


# Индексы на основе hash

- Подбирается функция перемешивания (hash function).
- Функция хеширования для ключа выдает номер участка.
- В памяти хранится таблица адресов участков.
- Доступ к данным по точному значению ключа возможен за одно обращение к диску.

# Пример: создание hash-индекса

```
CREATE INDEX hash_student_ind  
    USING HASH  
    ON STUDENT(StudentId)
```



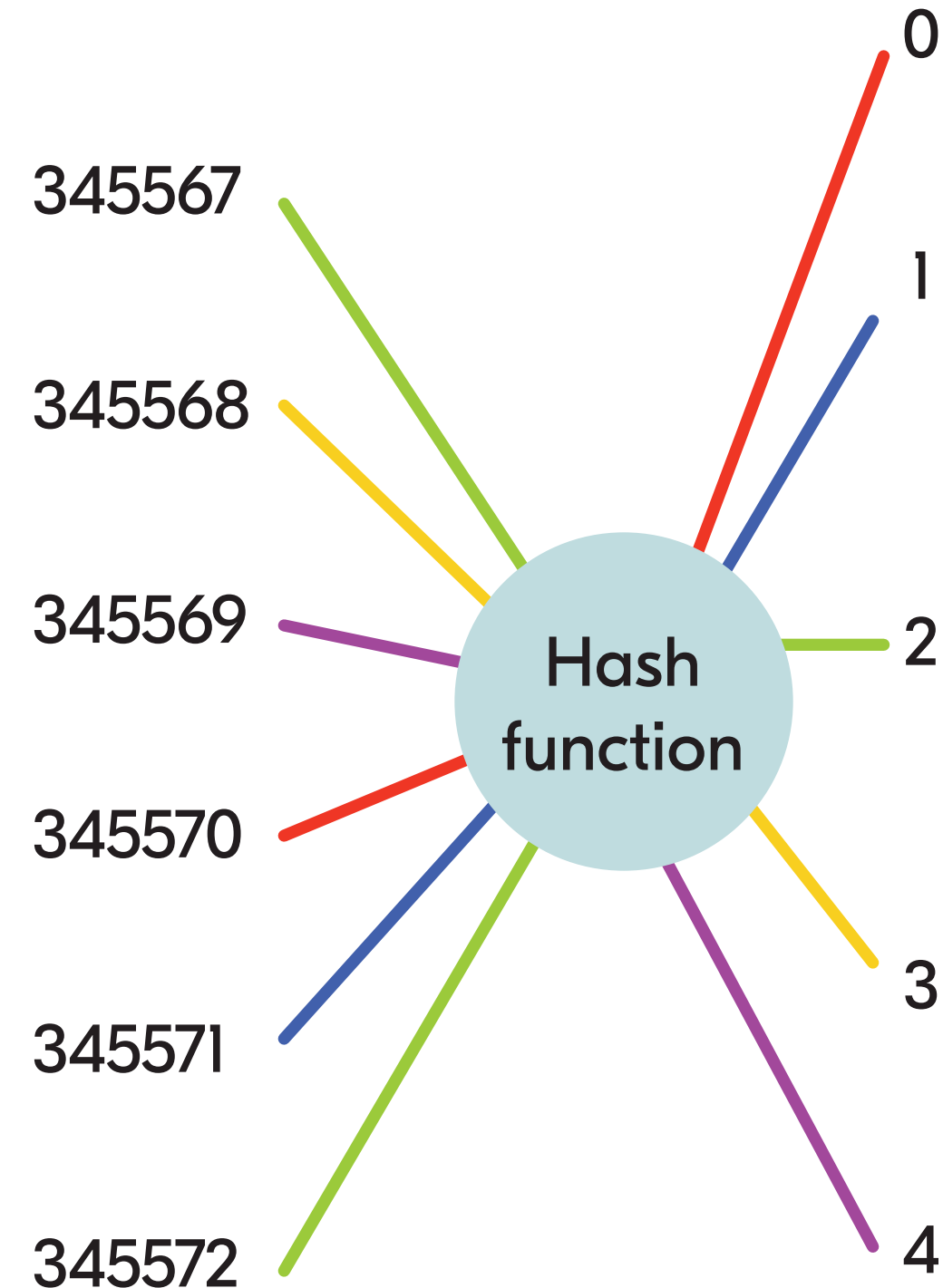
```
MySQL 5.7 Command Line Client  
line 1  
mysql> USE STUDENT_INFO;  
Database changed  
mysql> CREATE INDEX hash_student_ind USING HASH ON STUDENT(StudentId);  
Query OK, 0 rows affected (0.25 sec)  
Records: 0  Duplicates: 0  Warnings: 0  
  
mysql> _
```

# Пример: данные для построения hash-индекса

StudentId	StudentName	GroupNumber	BirthDate	Address
345567	Иванов Александр	341	1999-01-20	Петергоф, Библиотечная пл., дом 2
345568	Широков Федор	341	1998-03-21	Санкт-Петербург, ул. Гаванская 34
345569	Антонова Даша	341	1999-05-17	Санкт-Петербург, ул. Широкая 45
345570	Антипенко Денис	371	1998-08-11	Санкт-Петербург, Малый пр. 15
345571	Сидоров Александр	371	1999-07-12	Санкт-Петербург, Средний пр. 4
345572	Фадеев Дмитрий	371	1999-11-24	Санкт-Петербург, Невский пр. 23

# Пример: $\text{hash}(\text{key}) = \text{key} \bmod 5$

КЛЮЧИ

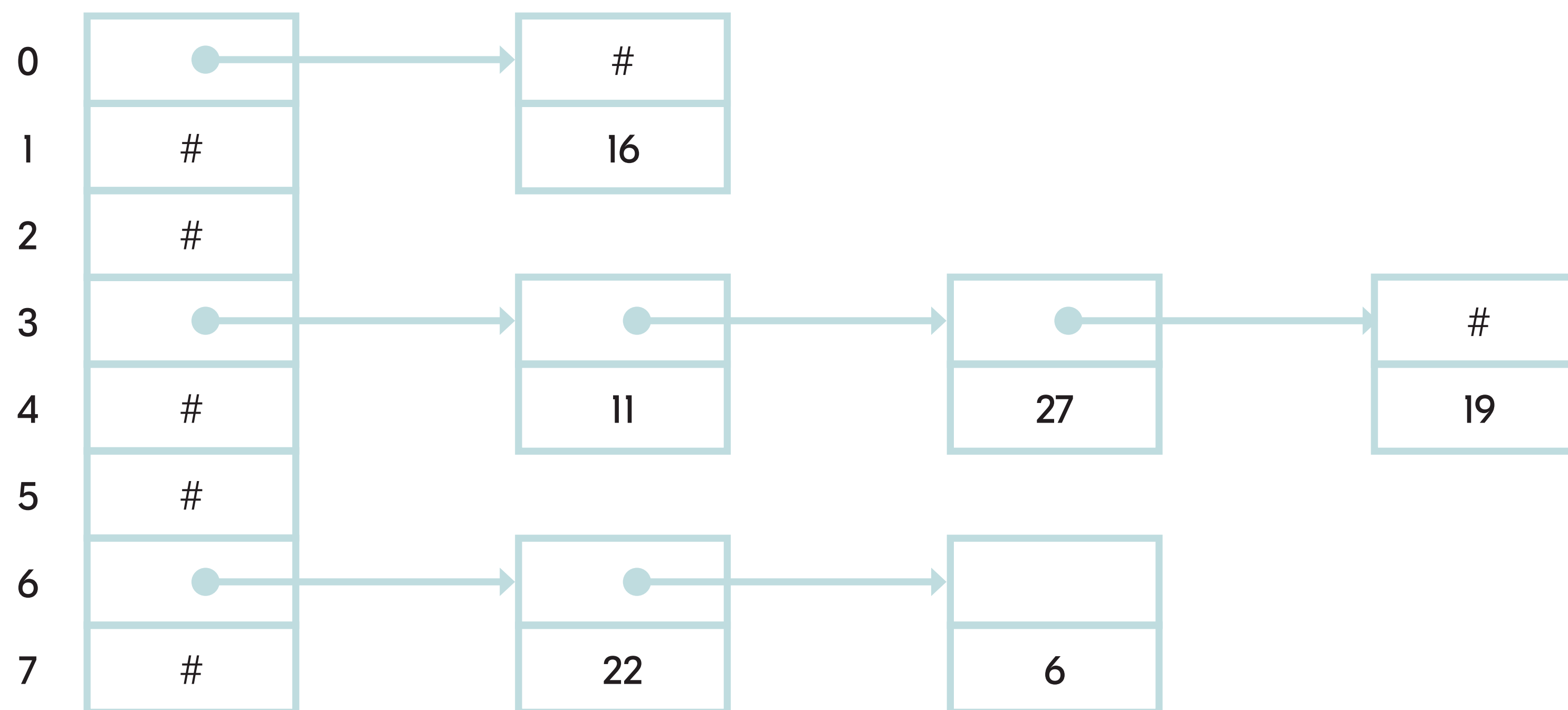


0	345570	Антипенко Денис	371	1998-08-11	Санкт-Петербург, Малый пр. 15
1	345571	Сидоров Александр	371	1999-07-12	Санкт-Петербург, Средний пр. 4
2	345567	Иванов Александр	341	1999-01-20	Петергоф, Библиотечная пл., дом 2
2	345572	Фадеев Дмитрий	371	1999-11-24	Санкт-Петербург, Невский пр. 23
3	345568	Широков Федор	341	1998-03-21	Санкт-Петербург, ул. Гаванская 34
4	345569	Антонова Даша	341	1999-05-17	Санкт-Петербург, ул. Широкая 45

# Недостатки hash-индексов

- Таблица адресов участков может не помещаться в оперативную память.
- Неравномерность размещения записей, возникновение коллизий.

# Пример: коллизии



# Индексы на основе битовых карт

- Подходят для столбцов с низкой избирательностью.
- Создаются быстро.
- Занимают мало места.
- Размер индекса на основе битовых карт существенно зависит от распределения данных.

# Построение индекса на основе битовых карт

- Для каждого значения индексируемого столбца — одна строка, состоящая из значения столбца и битовой последовательности.
- Длина битовой последовательности соответствует количеству строк таблицы.



# Пример: таблица Girls – данные с низкой избирательностью

Name	Eye_color	Hair_color	Height
Аня	карие	блондинка	средний
Даша	зеленые	блондинка	средний
Катя	голубые	шатенка	высокий
Таня	серые	рыжая	средний
Наташа	карие	брюнетка	средний
Марина	карие	блондинка	средний
Даша	серые	брюнетка	высокий
Оля	зеленые	шатенка	средний
Ира	голубые	рыжая	ниже среднего
Света	голубые	брюнетка	ниже среднего

# Пример: bitmap index на основе поля Height

- Высокий 0010001000
- Средний 1101110100
- Ниже среднего 0000000011

# Пример: ВІТМАР индекс на основе поля Hair\_color

- Блондинка 1100010000
- Шатенка 0010000100
- Брюнетка 0000101001
- Рыжая 0001000010

# Пример запроса: блондинка среднего роста

- Блондинка
  - Средний
  - Побитовое умножение
- $$\begin{array}{r} 1100010000 \\ \times 1101110100 \\ \hline 1100010000 \end{array}$$

Результат запроса:

- Аня
- Даша
- Марина.

# Пример: добавление нового значения в столбец Hair\_color

- Блондинка 11000100000
- Шатенка 00100001000
- Брюнетка 00001010010
- Рыжая 00010000100
- Голубые волосы 000000000001

# **Индексы на основе битовых карт**

Индексы на основе битовых карт удобны для поиска по нескольким столбцам с низкой избирательностью.