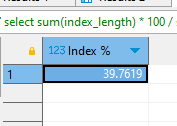
# **Задание 1**

Напишите запрос к учебной базе данных, который вернёт процентное отношение общего размера всех индексов к общему размеру всех таблиц.

# **Решение 1**

select sum(index\_length) \* 100 / sum(data\_length) as 'Index %'

from information\_schema.tables;



# **Задание 2**

Выполните explain analyze следующего запроса:

select distinct concat(c.last\_name, ' ', c.first\_name), sum(p.amount) over (partition by c.customer\_id, f.title)

from payment p, rental r, customer c, inventory i, film f

where date(p.payment\_date) = '2005-07-30' and p.payment\_date = r.rental\_date and r.customer\_id = c.customer\_id and i.inventory\_id = r.inventory\_id

* перечислите узкие места;
* оптимизируйте запрос: внесите корректировки по использованию операторов, при необходимости добавьте индексы.

# **Решение 2**

1. неявные объединения таблиц, лучше через join
2. sum over может сожрать много времени
3. как и distinct

**SELECT**

**CONCAT**(*c*.last\_name, **' '**, *c*.first\_name) **AS** *customer\_name*,

**SUM**(*p*.amount) **AS** *total\_amount*

**FROM**

sakila.payment *p*

**INNER** **JOIN** sakila.rental *r* **ON** *p*.payment\_date = *r*.rental\_date

**INNER** **JOIN** sakila.customer *c* **ON** *r*.customer\_id = *c*.customer\_id

**INNER** **JOIN** sakila.inventory *i* **ON** *r*.inventory\_id = *i*.inventory\_id

**INNER** **JOIN** sakila.film *f* **ON** *i*.film\_id = *f*.film\_id

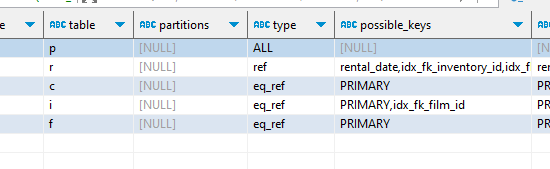
**WHERE**

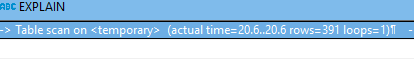
*p*.payment\_date >= **'2005-07-30'** **and** *p*.payment\_date < **DATE\_ADD**(**'2005-07-30'**, **INTERVAL** 1 **DAY**)

**GROUP** **BY**

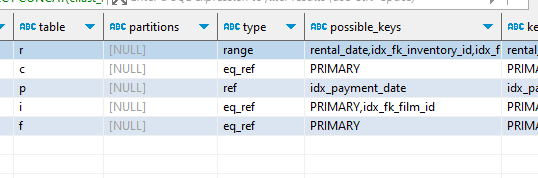
**CONCAT**(*c*.last\_name, **' '**, *c*.first\_name);

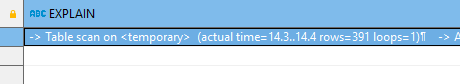
До создания индекса на payment\_date:





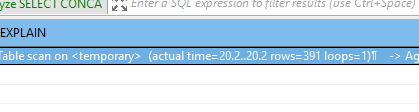
После создания индекса:



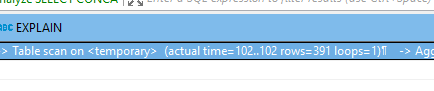


Разница есть, но несущественная.

Но если взять старую версию запроса с date() и прогнать ее без индекса



и с индексом



то видно, что его наличие даже вредит такой конструкции.