

Базы данных. SQL. Индексы.

Окулов Антон



R.class

DESCRIBE

```
-- Описание структуры таблицы
```

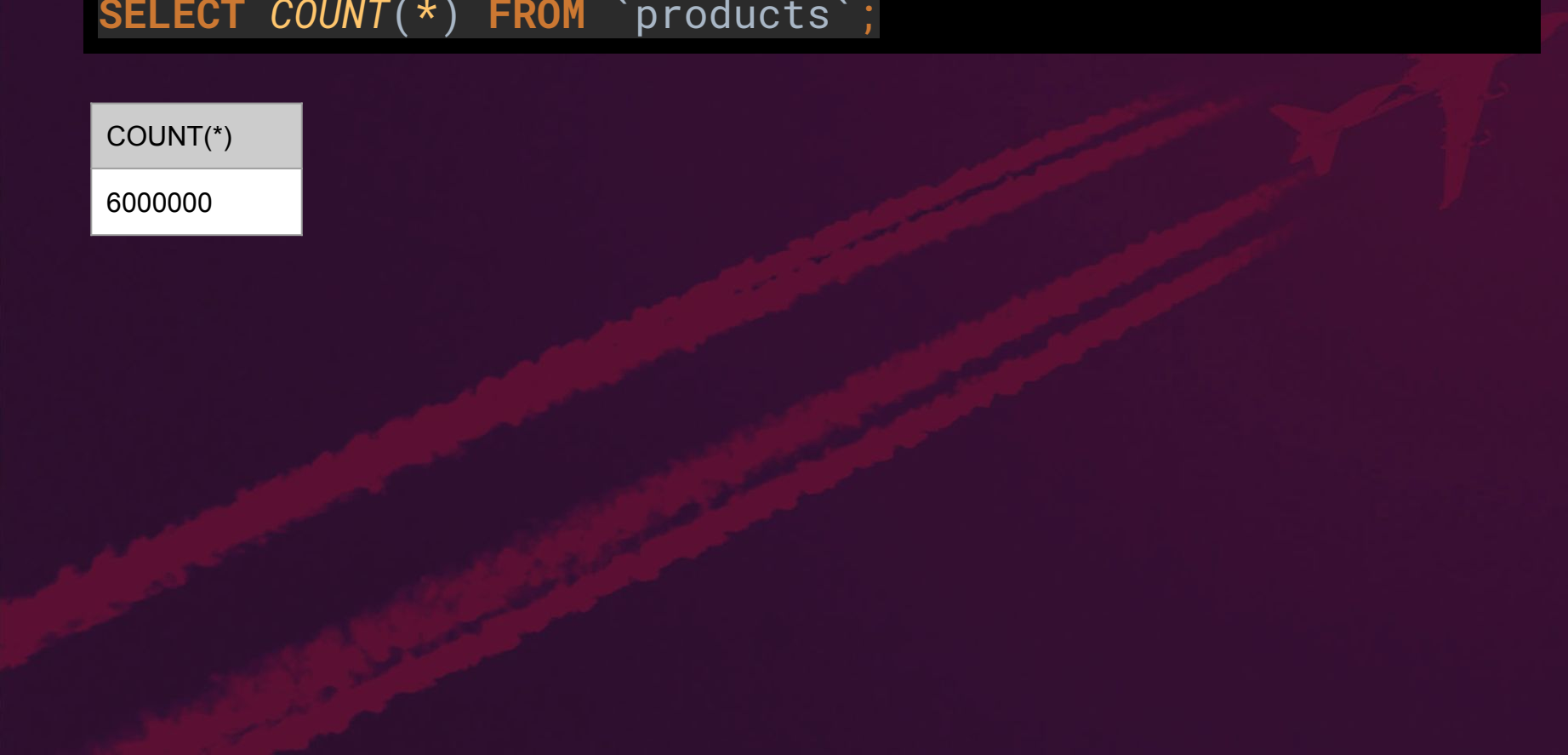
```
DESCRIBE `products`;
```

Field	Type	Null	Key	Default	Extra
id	int(11) unsigned	NO	PRI	NULL	auto_increment
price	int(11)	YES		NULL	
name	varchar(255)	YES		NULL	
mark	varchar(255)	YES		NULL	
vendor	varchar(255)	YES		NULL	
category	varchar(255)	YES		NULL	
year	year(4)	YES		NULL	

COUNT

```
SELECT COUNT(*) FROM `products`;
```

COUNT(*)
6000000



SELECT

```
SELECT * FROM products WHERE vendor = 'ZEUS' LIMIT 100 OFFSET 1500;
```



SELECT

```
SELECT * FROM products WHERE vendor = 'ZEUS' LIMIT 100 OFFSET 1500;
```



SELECT


```
SELECT * FROM products WHERE vendor = 'ZEUS' LIMIT 100 OFFSET 1500;
```

Время выполнения запроса	1.80s
--------------------------	-------

FULL SCAN


id	price	name	mark	vendor	category	year
1	5114	ZX - 339	EeMa6b0Zerh FRdp	Kindermann	Телефоны	2004
2	5774	z6 - 395	PSaRWggiOY bEsVf	iShoxs	Морозильники	1996
3	4951	2O - 239	uTXh9dBAtO U56uR	Urban Armor Gear	Вспышки	1990
4	8000	9i - 330	JeiQtS3cd3RF UBC	Sanitas	Игровые приставки	2020
...
5999999	7344	Bt - 363	gBkhDWHuog bHAh3	myOne	Мышки	1993
6000000	6708	H4 - 326	nNZGtQDRD WBkzXF	Partner (Instr)	Игры	1998

FULL SCAN



id	price	name	mark	vendor	category	year
1	5114	ZX - 339	EeMa6b0Zerh FRdp	Kindermann	Телефоны	2004
2	5774	z6 - 395	PSaRWggiOY bEsVf	iShoxs	Морозильники	1996
3	4951	2O - 239	uTXh9dBAtO U56uR	Urban Armor Gear	Вспышки	1990
4	8000	9i - 330	JeiQtS3cd3RF UBC	Sanitas	Игровые приставки	2020
...
5999999	7344	Bt - 363	gBkhDWHuog bHAh3	myOne	Мышки	1993
6000000	6708	H4 - 326	nNZGtQDRD WBkzXF	Partner (Instr)	Игры	1998

FULL SCAN



id	price	name	mark	vendor	category	year
1	5114	ZX - 339	EeMa6b0Zerh FRdp	Kindermann	Телефоны	2004
2	5774	z6 - 395	PSaRWggiOY bEsVf	iShoxs	Морозильники	1996
...	uTXh9dBAtO	Urban Armor
Количество проверок					6000000	
4	8000	9i - 330	JeIQtS3cd3RF UBC	Sanitas	Игровые приставки	2020
...
5999999	7344	Bt - 363	gBkhDWHuog bHAh3	myOne	Мышки	1993
6000000	6708	H4 - 326	nNZGtQDRD WBkzXF	Partner (Instr)	Игры	1998

COUNT vendor

```
SELECT COUNT(DISTINCT vendor) FROM products;
```

COUNT(DISTINCT vendor)


4085

INDEX

vendor
1C
1toy
1C-СофтКлаб
2DS/3DS
...
Эфир
Яркий Луч




INDEX



vendor
1C
1toy
1C-СофтКлаб
2DS/3DS
...
Эфир
Яркий Луч



INDEX



vendor
1C
1toy
1C События
2DS/3DS
...
Эфир
Яркий Луч

Количество проверок

4085



INDEX



vendor		
1C		
1toy		
1C Софт/КноБ		
Количество проверок	400	
2DS/3DS		
...		
Эфир		
Яркий Луч		

B-Tree Index

...

A

B

C

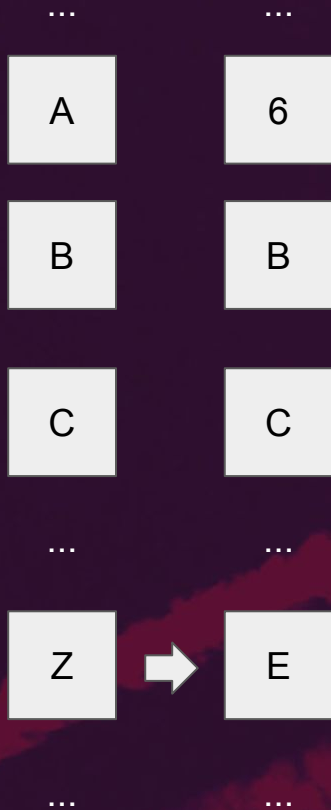
...

Z

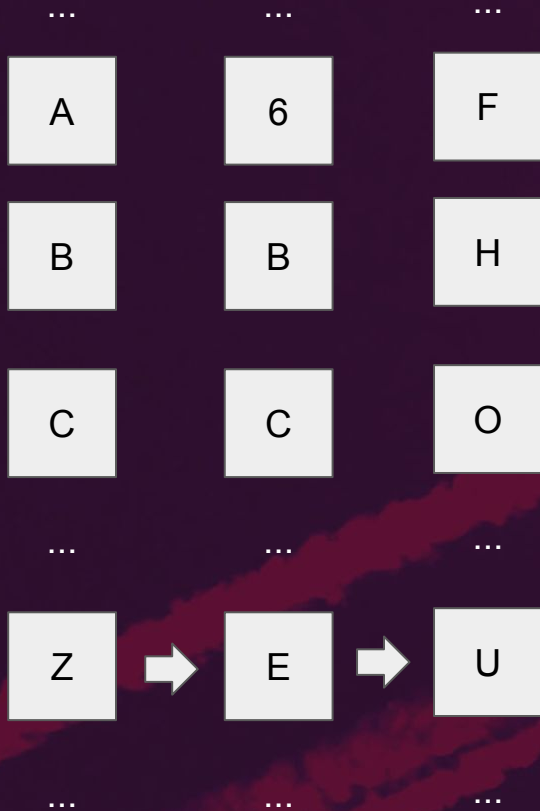
...



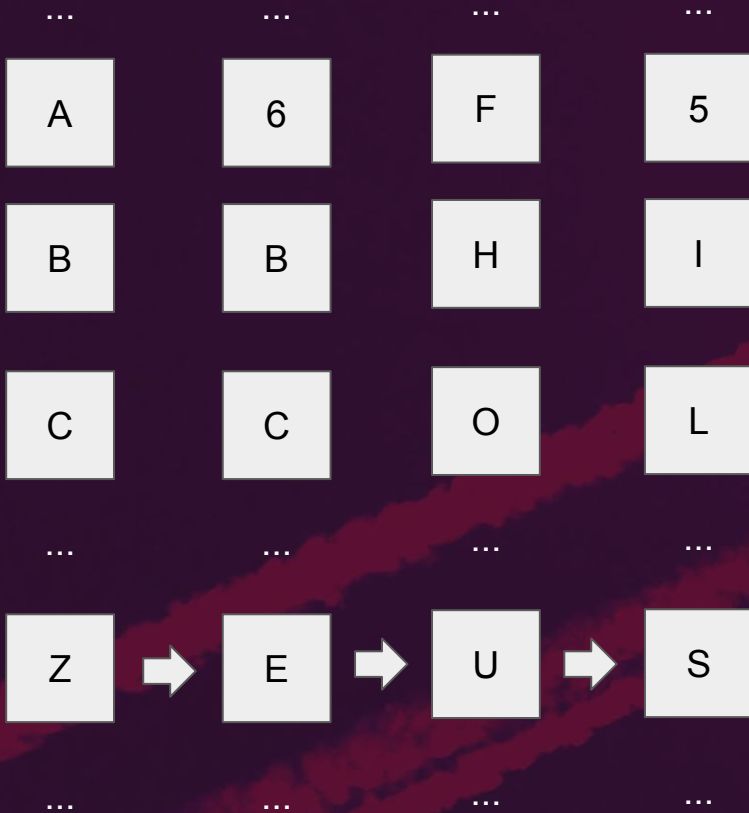
B-Tree Index



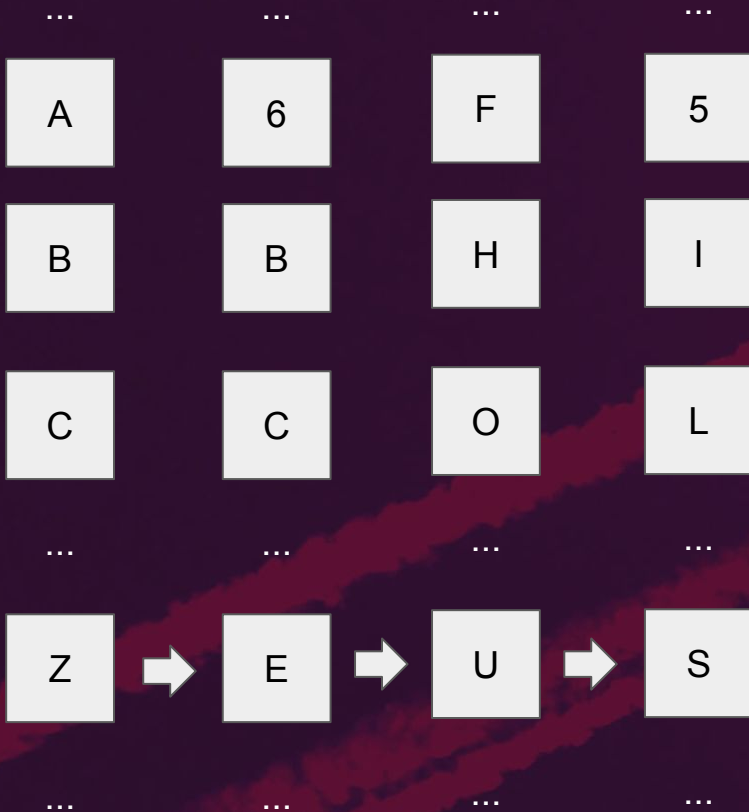
B-Tree Index



B-Tree Index

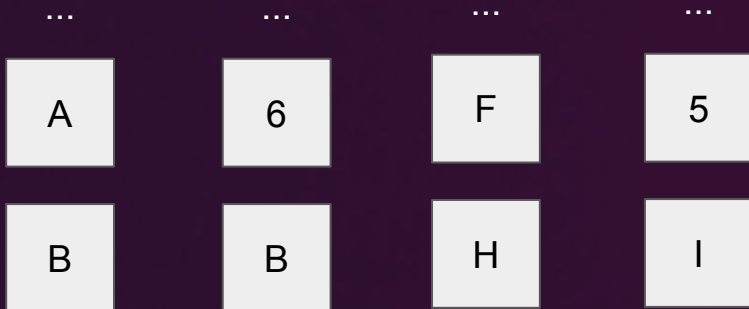


B-Tree Index



Этап поиска	Количество совпадений
Z___	59
ZE__	14
ZEU_	8
ZEUS	1

B-Tree Index

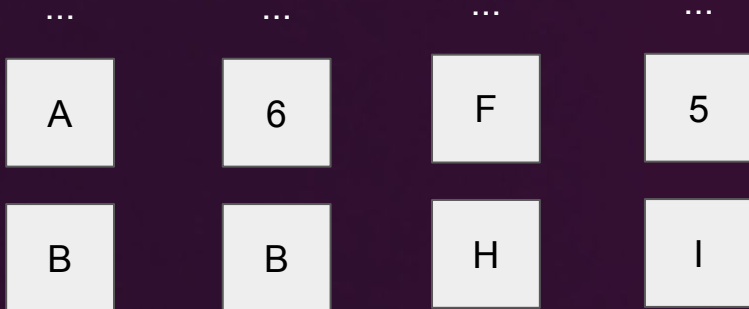


Количество проверок

82



B-Tree Index

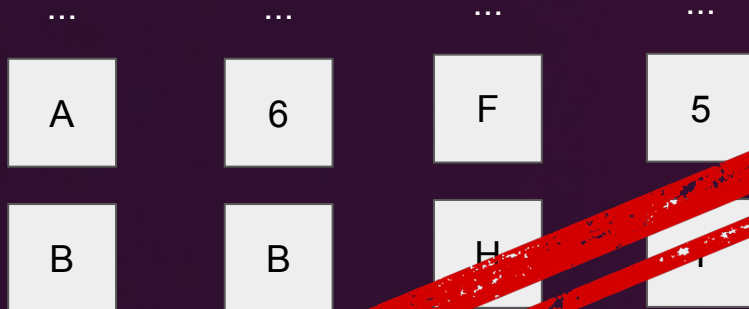


Ускорение в

73170 раз



B-Tree Index

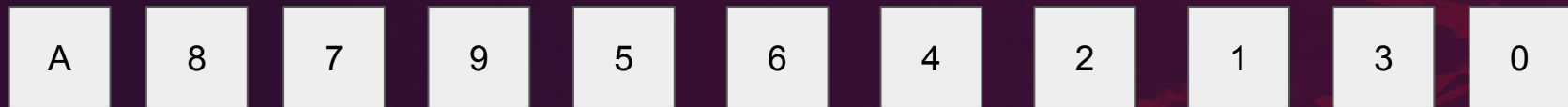


Ускорение поиска

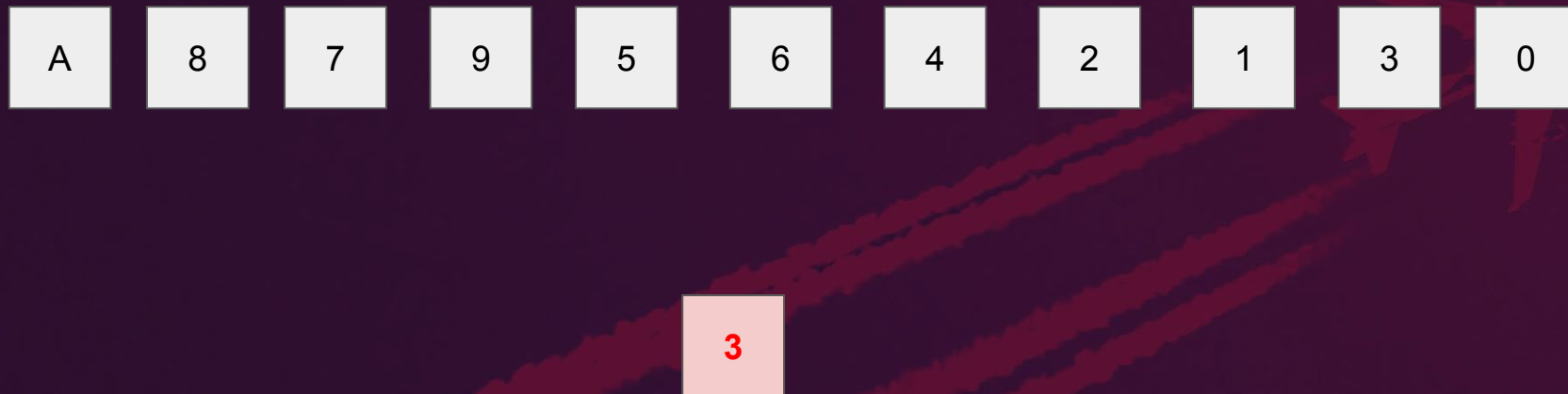
73170 раз



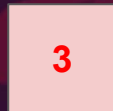
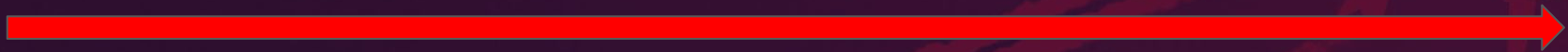
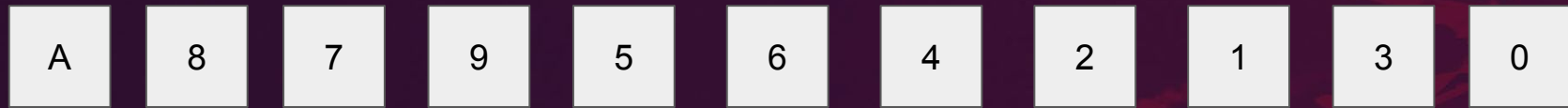
B-Tree Index



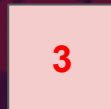
B-Tree Index



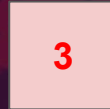
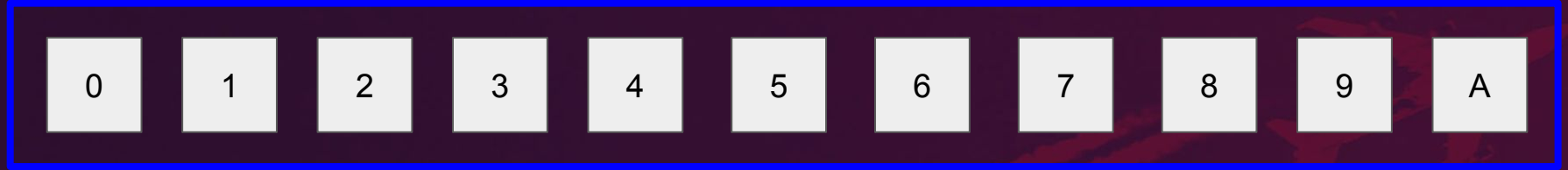
B-Tree Index



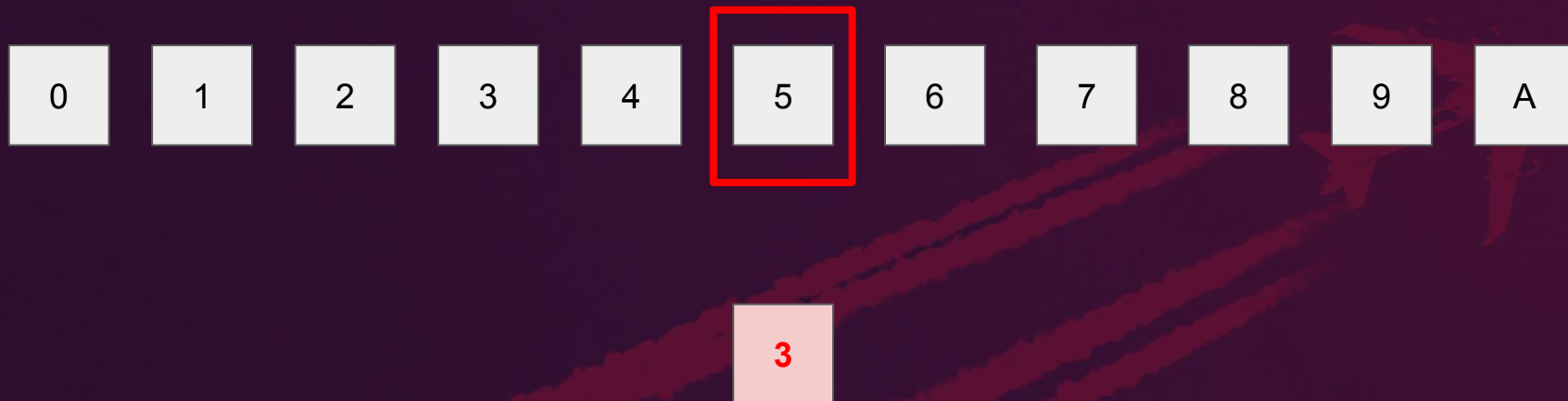
B-Tree Index



B-Tree Index



B-Tree Index



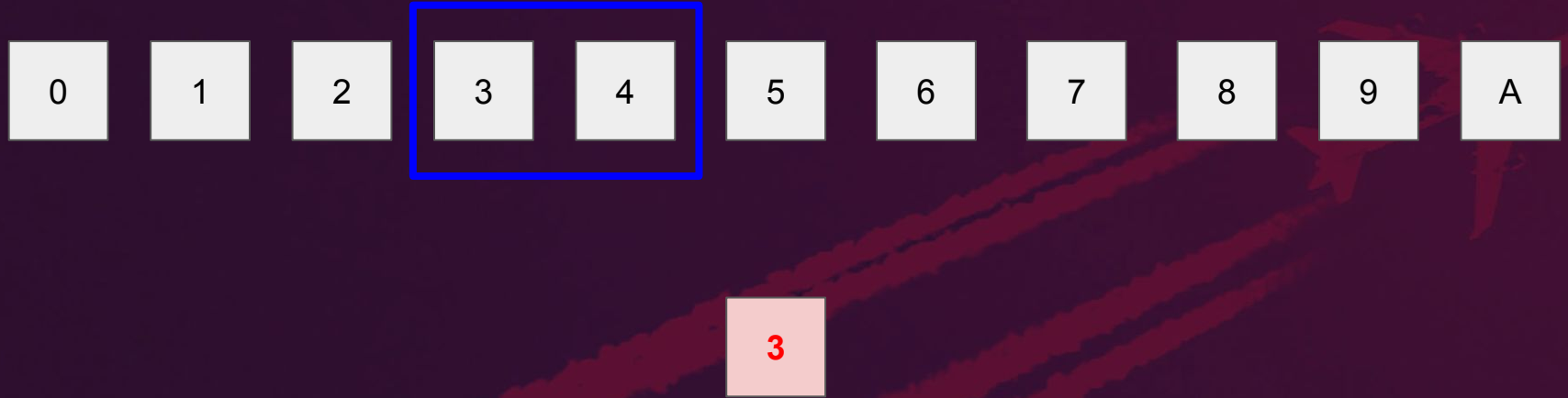
B-Tree Index



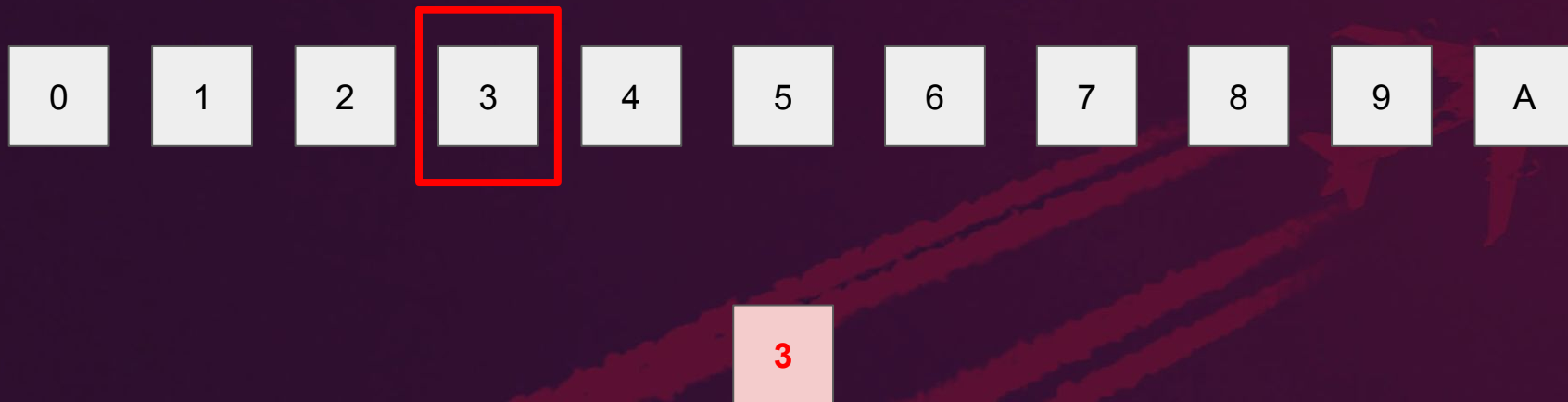
B-Tree Index



B-Tree Index



B-Tree Index



B-Tree Index



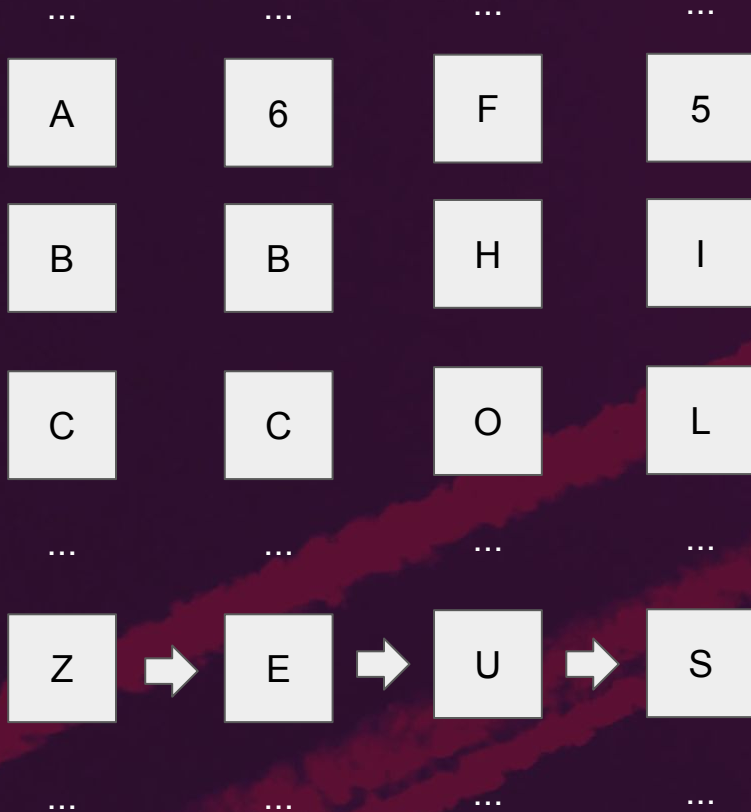
3 проверки вместо 12

B-Tree Index



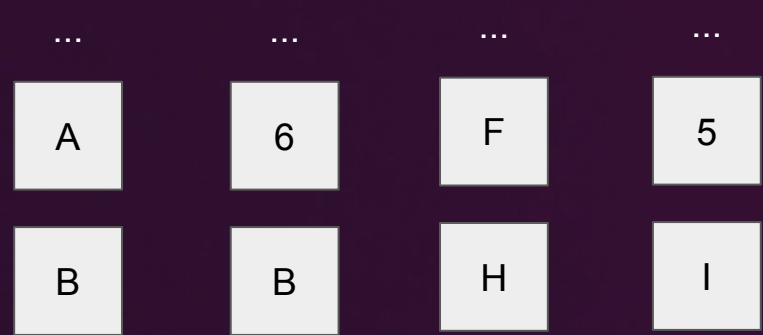
$$\log_2 n$$

B-Tree Index



Этап поиска	Количество совпадений	$\log_2 n$
Z__	59	6
ZE__	14	4
ZEU_	8	3
ZEUS	1	1

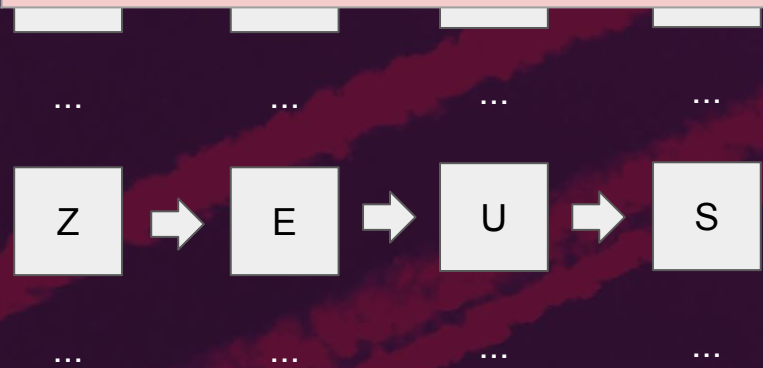
B-Tree Index



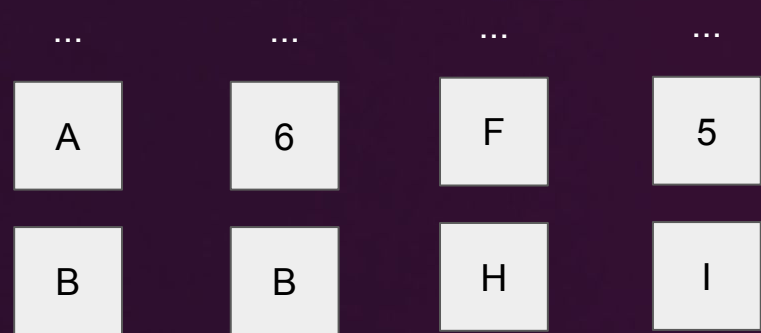
Этап поиска	Количество совпадений	$\log_2 n$
Z__	59	6
ZE__	14	4
ZEU	8	3
ZEUS	1	1

Количество проверок

14



B-Tree Index



Этап поиска	Количество совпадений	$\log_2 n$
Z__	59	6
ZE__	14	4
ZEU	8	3
ZEUS	1	1

Ускорение в 428571 раз

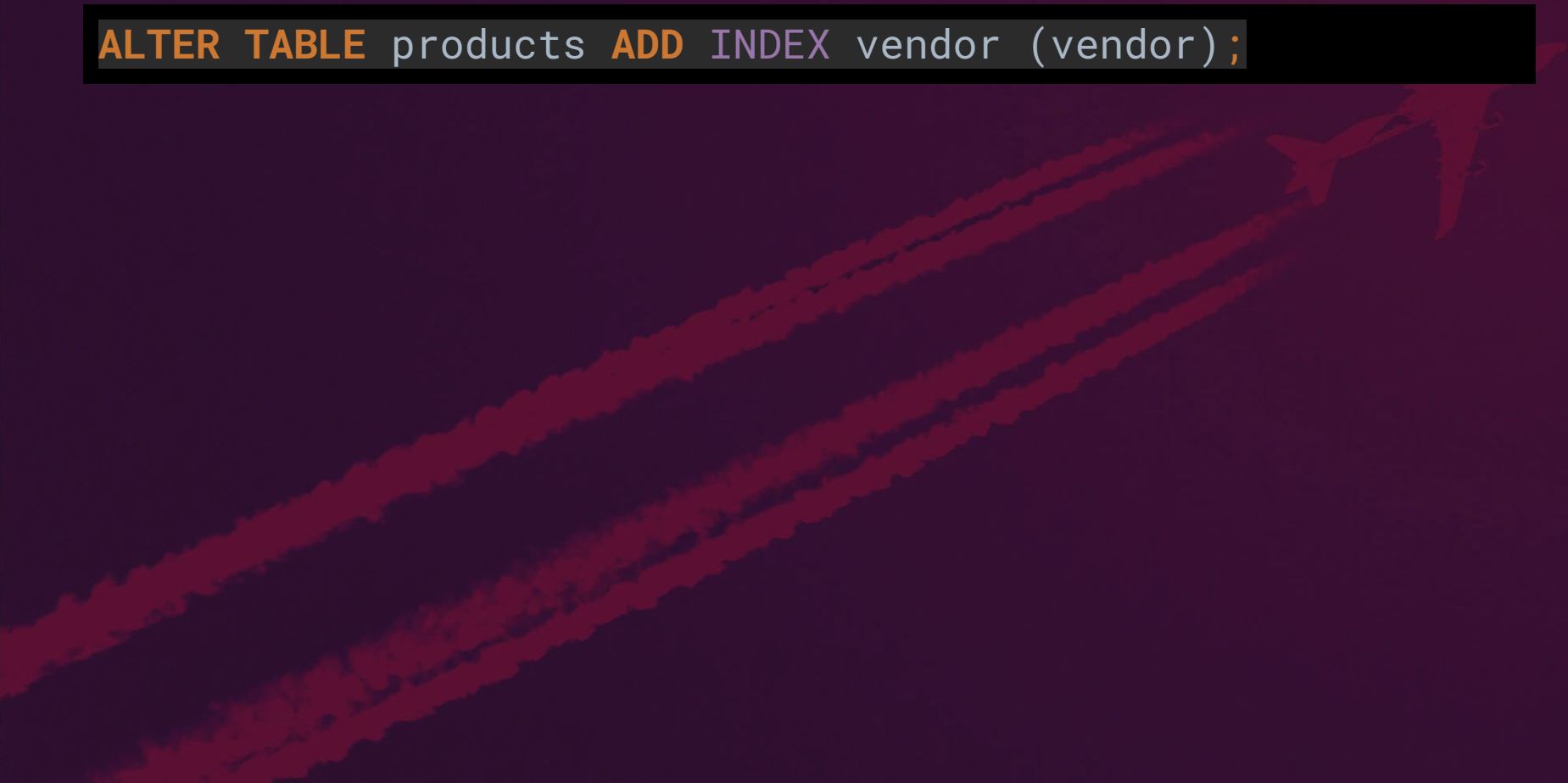


B-Tree Index

...	Этап	Количество совпадений	$\log_2 n$
A	6	F				6
B	B					4
						3
Ускорение в					8571 раз	
						1
...			
Z	→ E	→ U				
...			

Добавим индекс

```
ALTER TABLE products ADD INDEX vendor (vendor);
```



SELECT

```
SELECT * FROM products WHERE vendor = 'ZEUS' LIMIT 100 OFFSET 1500;
```



SELECT

```
SELECT * FROM products WHERE vendor = 'ZEUS' LIMIT 100 OFFSET 1500;
```

Время выполнения запроса	37ms
--------------------------	------

Просмотр индексов

```
SHOW INDEX FROM products;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Cardinality	Null	Index_type
products	0	PRIMARY	1	id	5808034		BTREE
products	1	vendor	1	vendor	4086	YES	BTREE

SELECT

```
SELECT * FROM products  
WHERE price = 9990 and category = 'Телевизоры';
```

Время выполнения запроса	1.69s
--------------------------	-------

Добавим индекс

```
ALTER TABLE products
```

```
ADD INDEX price_category (price, category);
```

Добавим индекс

```
ALTER TABLE products  
ADD INDEX price_category (price, category);
```

VS

```
ALTER TABLE products  
ADD INDEX category_price (category, price);
```

COUNT price

```
SELECT COUNT(*) FROM products GROUP BY price;
```

COUNT(*)
612
552
604
596
...

COUNT category

```
SELECT COUNT(*) FROM products GROUP BY category;
```

COUNT(*)
260750
260977
262209
259792
...

Добавим индекс

```
ALTER TABLE products  
ADD INDEX price_category (price, category);
```

VS

```
ALTER TABLE products  
ADD INDEX category_price (category, price);
```


Добавим индекс

```
ALTER TABLE products
```

```
ADD INDEX price_category (price, category);
```

Время выполнения запроса

0.8ms

```
ALTER TABLE products
```

```
ADD INDEX category_price (category, price);
```

Время выполнения запроса

0.9ms

Селективность индекса

Вариант price_category:

- После первой части индекса число записей сократится до 590-600
- Останется сделать выборку из них

Вариант category_price:

- После первой части индекса число записей сократится до 260 тысяч
- Нужно будет сделать выборку из них

Чем меньшему количеству строк соответствует значение атрибута - тем выше селективность. Такие атрибуты следует использовать в начале индекса.

Просмотр индексов

```
SHOW INDEX FROM products;
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Cardinality	Null	Index_type
products	0	PRIMARY	1	id	5808034		BTREE
products	1	vendor	1	vendor	4086	YES	BTREE
products	1	price_category	1	price	12297	YES	BTREE
products	1	price_category	2	category	234304	YES	BTREE

SELECT

```
SELECT * FROM `products`  
WHERE `price` = 9990 and `category` = 'Телевизоры'  
ORDER BY `year`;
```

SELECT

```
SELECT * FROM `products`  
WHERE `price` = 9990 and `category` = 'Телевизоры'  
ORDER BY `year`;
```

Атрибуты в ORDER BY необходимо добавлять в “хвост” составного индекса

SELECT

```
SELECT * FROM `products`  
WHERE `price` = 9990 and `category` = 'Телевизоры'  
ORDER BY `year`;
```

Атрибуты в ORDER BY необходимо добавлять в “хвост” составного индекса

```
ALTER TABLE products  
ADD INDEX price_category_year (price, category, year);
```

EXPLAIN

```
EXPLAIN SELECT * FROM products  
WHERE price = 9990 and category = 'Телевизоры';
```

EXPLAIN до добавления индекса

```
EXPLAIN SELECT * FROM products  
WHERE price = 9990 and category = 'Телевизоры';
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows
1	SIMPLE	products		ALL					5820165

EXPLAIN после добавления индекса

```
EXPLAIN SELECT * FROM products  
WHERE price = 9990 and category = 'Телевизоры';
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows
1	SIMPLE	products		ref	price_category	price_category	773	const,const	28

EXPLAIN после добавления индекса

```
EXPLAIN SELECT * FROM products  
WHERE `price` = 9990 and year = 2000;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows
1	SIMPLE	products		ref	price_category	price_category	5	const	617

Расчет длины ключа

Varchar:

$$255 * 3 + 2 + 1 = 768$$

(Длина Varchar) * (Максимальное число байт на символ в кодировке) + 2
(Переменная длина строки) + 1 (Потому что используем NULL)

Int:

$$4 + 1 = 5$$

(Int - 4 байта) + 1 (Потому что используем NULL)

Общее число используемых байт в ключе: 773

Предостережения

- Не нужно заранее создавать индексы
- Удаляйте неиспользуемые индексы
- Не используйте индексы на небольших (до нескольких тысяч записей) таблицах
- Исходите из медленных запросов - создавайте уникальные индексы под них
- Перед созданием индексов в проде - крайне желательно проверить это на локальной копии данных



Прочитать и изучить

<https://ruhighload.com/%D0%98%D0%BD%D0%B4%D0%B5%D0%BA%D1%81%D1%8B+%D0%B2+mysql> - ОБЯЗАТЕЛЬНО

http://highload.guide/blog/basics_indexing.html - КРАЙНЕ ЖЕЛАТЕЛЬНО

<https://ru.wikipedia.org/wiki/B-%D0%B4%D0%B5%D1%80%D0%B5%D0%B2%D0%BE>

<https://habr.com/ru/post/141767/>

<https://dev.mysql.com/doc/refman/8.0/en/explain.html>

<https://dev.mysql.com/doc/refman/8.0/en/explain-output.html>

Почитать и попробовать EXPLAIN ANALYZE

Bcë :)

