

Database Indexes for Fun and Speed

Imagine An Unsorted Phonebook

Row	Last Name	First Name	Middle Initial
1	Miller	Jessica	E
2	Grant	Anna	C
3	White	Beryl	D
4	Williams	Steven	R
5	Grant	Charlotte	A
6	Whitney	Ian	M
7	White	Arthur	D

Let's query that Phonebook

```
select * from phone_book where last_name = 'Whitney'
```

Let's add an index

```
create index phone_book_last_name on phone_book(last_name);
```

Last Name	Row
Grant	2
Grant	5
Miller	1
White	3
White	7
Whitney	6
Williams	4

But Looking By Last Name Sucks

```
select * from phone_book where last_name = 'Whitney' and first_name = 'Ian';
```

We could add another index

```
create index phone_book_first_name on phone_book(first_name);
```

First Name	Row
Anna	2
Arthur	7
Beryl	3
Charlotte	5
Ian	6
Jessica	1
Steven	4

This is probably better

But...

- Your database now has to maintain two indexes
- And none of your users ever query by just first name
 - It's always last name & first name

Let's add a better index

```
create index phone_book_first_last_name on phone_book(first_name, last_name);
```

Last Name	First Name	Row
Grant	Anna	2
Grant	Charlotte	5
Miller	Jessica	1
White	Arthur	7
White	Beryl	3
Whitney	Ian	6
Williams	Steven	4

Which Index Gets Used?

We now have 3 indexes. Which of them get used?

Query by lastname, firstname

```
select * from phone_book where last_name = 'Whitney' and first_name = 'Ian';
```

Uses: phone_book_first_last_name

Query just by first name

```
select * from phone_book where first_name = 'Ian';
```

Uses: phone_book_first_name

Query just by last name

```
select * from phone_book where last_name = 'Whitney'
```

Uses: phone_book_first_last_name

Delete that first name index!

It's redundant

More: <https://z.umn.edu/indexes>