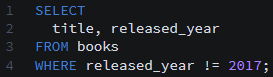
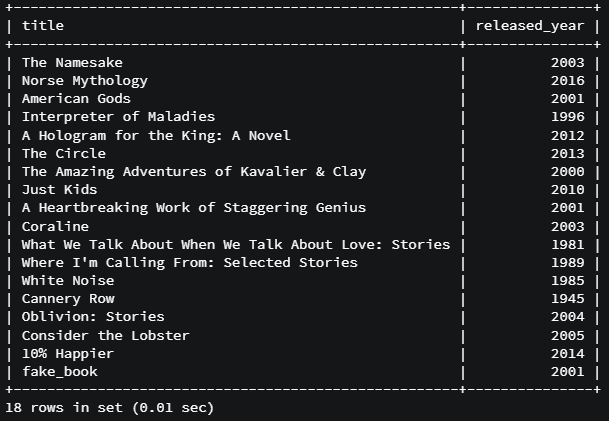
Section slides: <http://webdev.slides.com/coltsteele/mysql-99-103>

* Operators allow us to further refine our selections by narrowing down the selection criteria in specific ways
* We’ll do things like select birthdays between two defined years, or selecting all items in stock and priced below a certain price

# NOT EQUAL

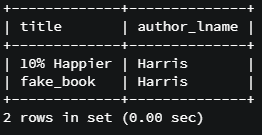
* The **not equal** operator, **“!=”**, is used to *exclude* the defined values
  + A fairly univers
* Documetation: <https://www.w3resource.com/mysql/comparision-functions-and-operators/not-equal-operator.php>
* It’s very similar to using the “=” operator, but it does the opposite
* Let’s seen an example where we *select the titles of all books that were NOT released in 2017*





* Another example where we exclude particular authors
  + To motivate the idea, we first select all books by authors with the last name “Harris”





* + Now we simply flip the switch and use the NOT EQUAL operator to select titles if books by all authors except those with the last name “Harris”





* Code summary

SELECT title FROM books WHERE released\_year = 2017;

SELECT title FROM books WHERE released\_year != 2017;

SELECT title, author\_lname FROM books;

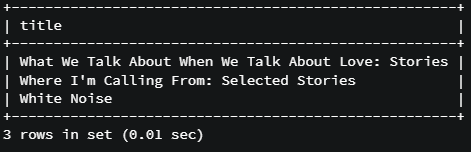
SELECT title, author\_lname FROM books WHERE author\_lname = 'Harris';

SELECT title, author\_lname FROM books WHERE author\_lname != 'Harris';

# NOT LIKE

* The NOT LIKE function is the opposite of LIKE. Recall that “LIKE” allows us to match patterns, usually in strings
* As a refresher, let’s do a LIKE selection for all books that start with “W”





* Now we’ll try a NOT LIKE selection. Here, we’ll select all titles that DO NOT start with the letter “W”
  + Notice how this does not use the exclamation point operator. We simply say “NOT LIKE”





* Code summary

SELECT title FROM books WHERE title LIKE 'W';

SELECT title FROM books WHERE title LIKE 'W%';

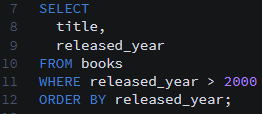
SELECT title FROM books WHERE title LIKE '%W%';

SELECT title FROM books WHERE title LIKE 'W%';

SELECT title FROM books WHERE title NOT LIKE 'W%';

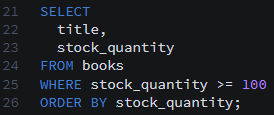
# GREATER THAN and GREATER THAN OR EQUAL TO

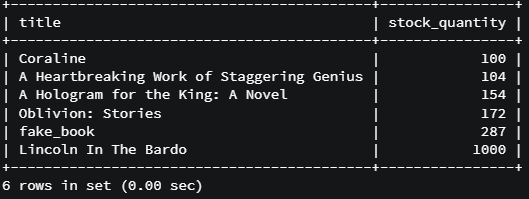
* The greater than operator is the “>” symbol. It is also fairly universal across programming languages
* With this operator, we can do things like *select all books released after the year 2000*





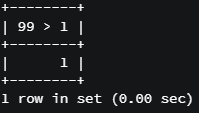
* The **greater than or equal to** operator, represented by “>=”, is inclusive of the defining value.
* Let’s do an example where we want to *select all books with stock of greater than or equal to 100*





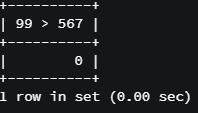
* What happens if we try the following?





* + What returns is the Boolean value of 1, representing “True”. Since 99 is in fact greater than 1, this selection will return “1”, which is equivalent to “True”.
  + To verify, let’s write a false statement. This will return “0”, equivalent to “False”





* Exercise: What will these evaluate to?
  + 100 > 5
    - 1 (True)
  + -15 > 15
    - 0 (False)
  + 9 > -10
    - 1 (True)
  + 1 > 1
    - 0 (False)
  + ‘a’ > ‘b’
    - 0 (False)
  + ‘A’ > ‘a’
    - 0 (False)
    - A and a are actually equivalent in MySQL. We’ve seen this in instances were we select text in a case-insensitive manner
* It is best to avoid logical operators on strings because it is confusing and inconsistent between different languages
* Code summary

SELECT title, released\_year FROM books ORDER BY released\_year;

SELECT title, released\_year FROM books

WHERE released\_year > 2000 ORDER BY released\_year;

SELECT title, released\_year FROM books

WHERE released\_year >= 2000 ORDER BY released\_year;

SELECT title, stock\_quantity FROM books;

SELECT title, stock\_quantity FROM books WHERE stock\_quantity >= 100;

SELECT 99 > 1;

SELECT 99 > 567;

100 > 5

-- true

-15 > 15

-- false

9 > -10

-- true

1 > 1

-- false

'a' > 'b'

-- false

'A' > 'a'

-- false

'A' >= 'a'

-- true

SELECT title, author\_lname FROM books WHERE author\_lname = 'Eggers';

SELECT title, author\_lname FROM books WHERE author\_lname = 'eggers';

SELECT title, author\_lname FROM books WHERE author\_lname = 'eGGers';