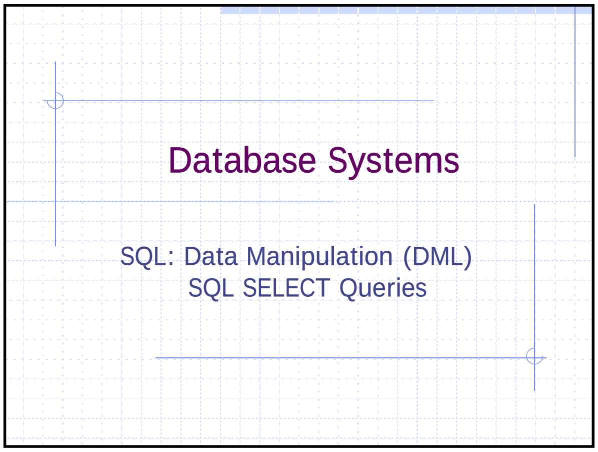
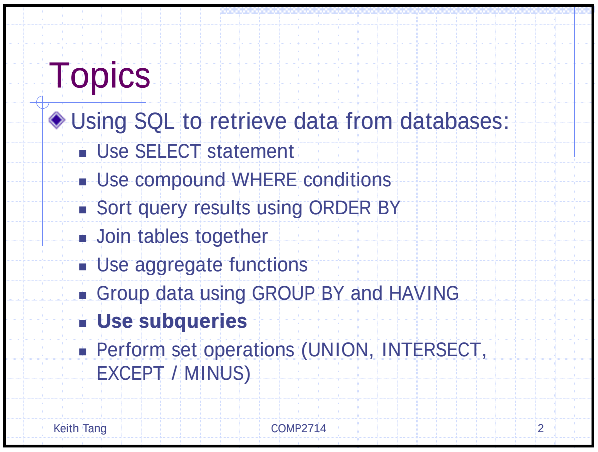


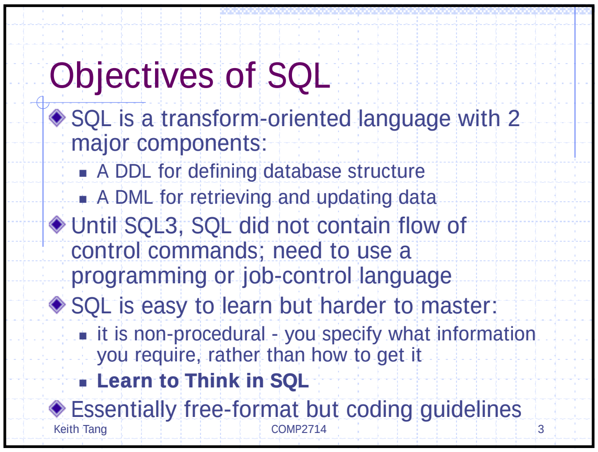
Database Systems

SQL: Data Manipulation (DML)
SQL SELECT Queries



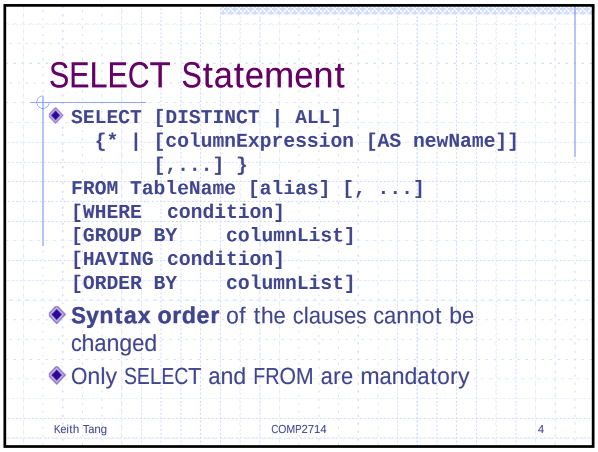
Topics

- Using SQL to retrieve data from databases:
  - Use SELECT statement
  - Use compound WHERE conditions
  - Sort query results using ORDER BY
  - Join tables together
  - Use aggregate functions
  - Group data using GROUP BY and HAVING
  - **Use subqueries**
  - Perform set operations (UNION, INTERSECT, EXCEPT / MINUS)



Objectives of SQL

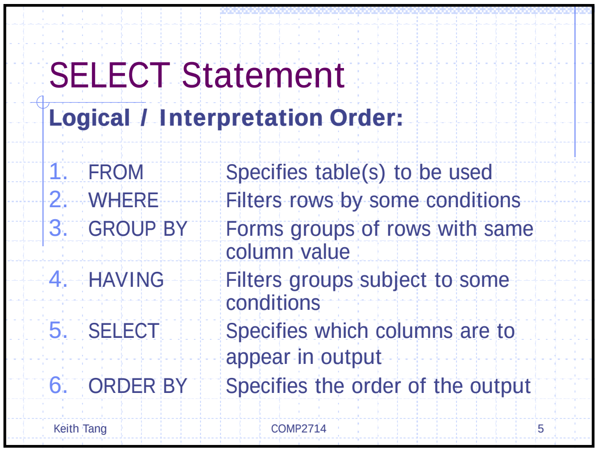
- SQL is a transform-oriented language with 2 major components:
  - A DDL for defining database structure
  - A DML for retrieving and updating data
- Until SQL3, SQL did not contain flow of control commands; need to use a programming or job-control language
- SQL is easy to learn but harder to master:
  - it is non-procedural - you specify what information you require, rather than how to get it
  - **Learn to Think in SQL**
- Essentially free-format but coding guidelines



SELECT Statement

```
SELECT [DISTINCT | ALL]
  {* | [columnExpression [AS newName]]
       [,...] }
FROM TableName [alias] [, ...]
[WHERE  condition]
[GROUP BY    columnList]
[HAVING condition]
[ORDER BY    columnList]
```

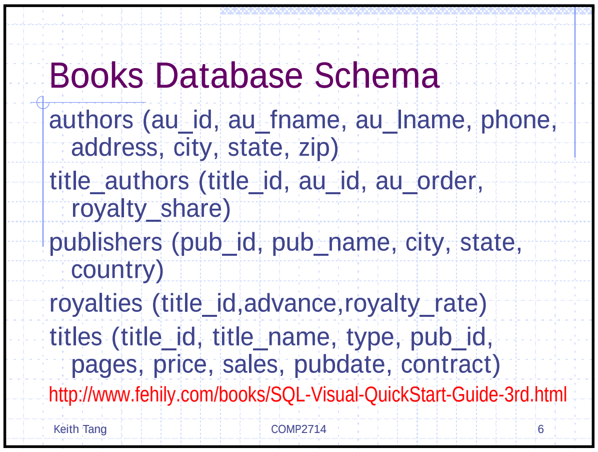
- **Syntax order** of the clauses cannot be changed
- Only SELECT and FROM are mandatory



SELECT Statement

Logical / Interpretation Order:

1.	FROM	Specifies table(s) to be used
2.	WHERE	Filters rows by some conditions
3.	GROUP BY	Forms groups of rows with same column value
4.	HAVING	Filters groups subject to some conditions
5.	SELECT	Specifies which columns are to appear in output
6.	ORDER BY	Specifies the order of the output



Books Database Schema

authors (au_id, au_fname, au_lname, phone, address, city, state, zip)

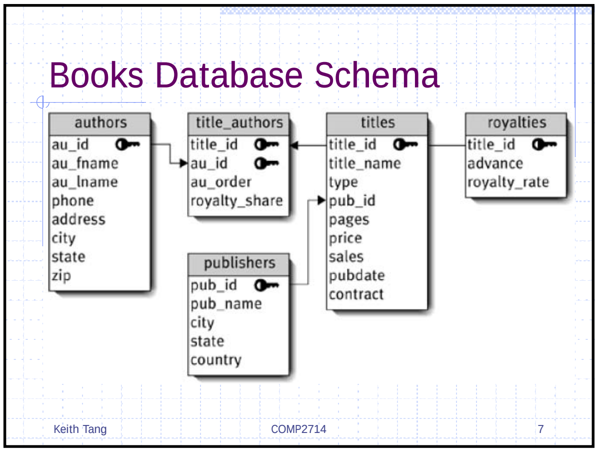
title_authors (title_id, au_id, au_order, royalty_share)

publishers (pub_id, pub_name, city, state, country)

royalties (title_id,advance,royalty_rate)

titles (title_id, title_name, type, pub_id, pages, price, sales, pubdate, contract)

http://www.fehily.com/books/SQL-Visual-QuickStart-Guide-3rd.html



Books Database Schema

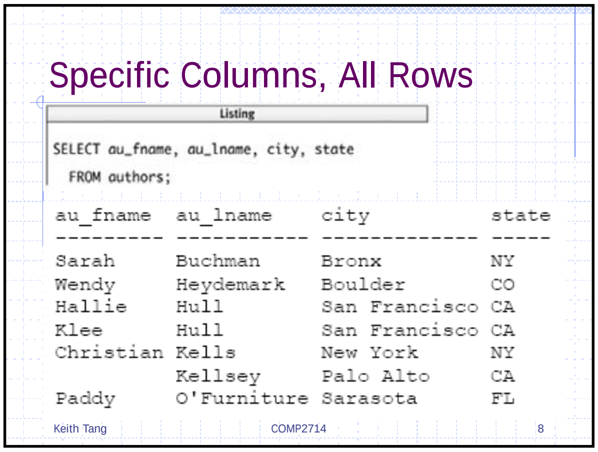
authors
au_id
au_fname
au_lname
phone
address
city
state
zip

title_authors
title_id
au_id
au_order
royalty_share

titles
title_id
title_name
type
pub_id
pages
price
sales
pubdate
contract

royalties
title_id
advance
royalty_rate

publishers
pub_id
pub_name
city
state
country

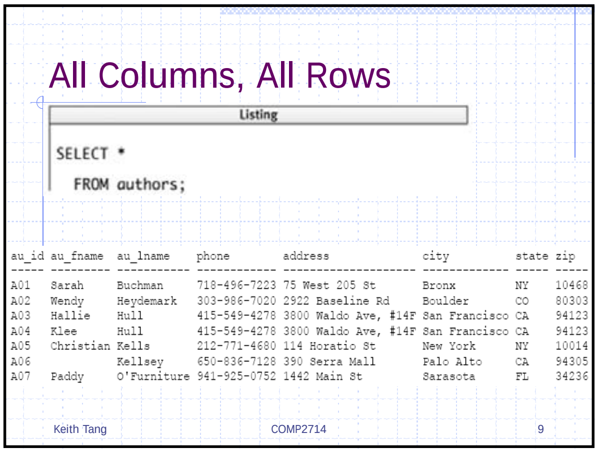


Specific Columns, All Rows

Listing

```
SELECT au_fname, au_lname, city, state
  FROM authors;
```

```
au_fname  au_lname    city          state
--------- ----------- ------------- -----
Sarah     Buchman     Bronx         NY
Wendy     Heydemark   Boulder       CO
Hallie    Hull        San Francisco CA
Klee      Hull        San Francisco CA
Christian Kells       New York      NY
          Kellsey     Palo Alto     CA
Paddy     O'Furniture Sarasota      FL
```

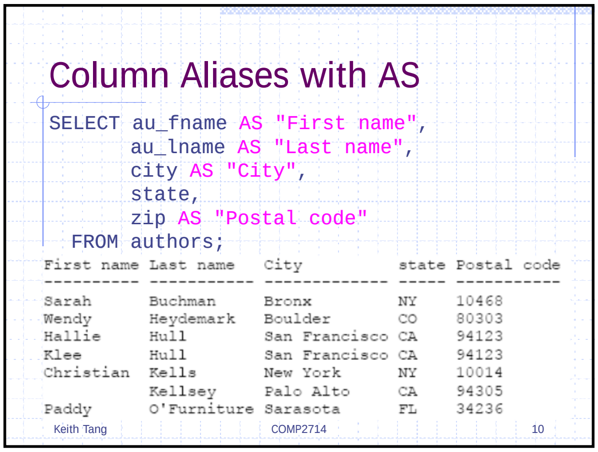


All Columns, All Rows

Listing

```
SELECT *
  FROM authors;
```

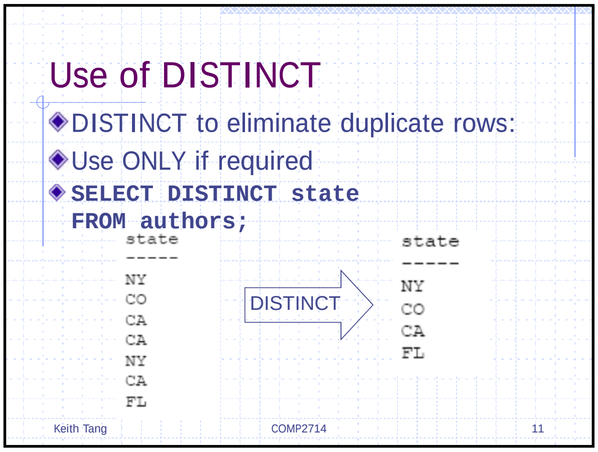
```
au_id au_fname  au_lname    phone        address              city          state zip
----- --------- ----------- ------------ -------------------- ------------- ----- -----
A01   Sarah     Buchman     718-496-7223 75 West 205 St       Bronx         NY    10468
A02   Wendy     Heydemark   303-986-7020 2922 Baseline Rd     Boulder       CO    80303
A03   Hallie    Hull        415-549-4278 3800 Waldo Ave, #14F San Francisco CA    94123
A04   Klee      Hull        415-549-4278 3800 Waldo Ave, #14F San Francisco CA    94123
A05   Christian Kells       212-771-4680 114 Horatio St       New York      NY    10014
A06             Kellsey     650-836-7128 390 Serra Mall       Palo Alto     CA    94305
A07   Paddy     O'Furniture 941-925-0752 1442 Main St         Sarasota      FL    34236
```



Column Aliases with AS

```
SELECT au_fname AS "First name",
       au_lname AS "Last name",
       city AS "City",
       state,
       zip AS "Postal code"
  FROM authors;
```

```
First name Last name   City          state Postal code
---------- ----------- ------------- ----- -----------
Sarah      Buchman     Bronx         NY    10468
Wendy      Heydemark   Boulder       CO    80303
Hallie     Hull        San Francisco CA    94123
Klee       Hull        San Francisco CA    94123
Christian  Kells       New York      NY    10014
           Kellsey     Palo Alto     CA    94305
Paddy      O'Furniture Sarasota      FL    34236
```



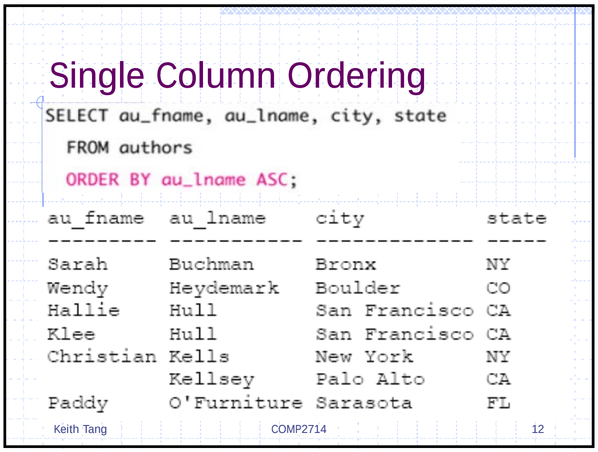
Use of DISTINCT

- DISTINCT to eliminate duplicate rows:
- Use ONLY if required
- **SELECT DISTINCT state FROM authors;**

```
state
-----
NY
CO
CA
CA
NY
CA
FL
```

DISTINCT

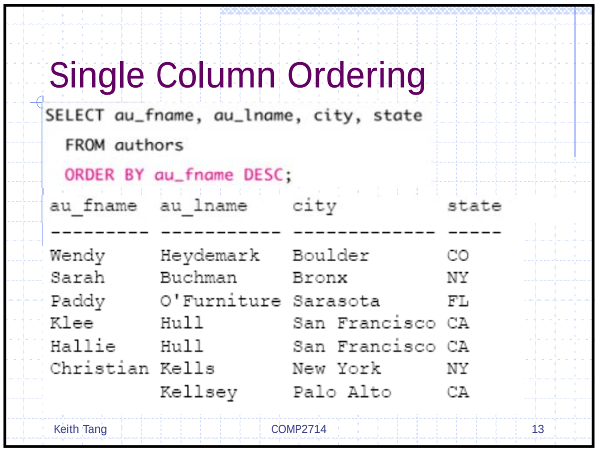
```
state
-----
NY
CO
CA
FL
```



Single Column Ordering

```
SELECT au_fname, au_lname, city, state
  FROM authors
  ORDER BY au_lname ASC;
```

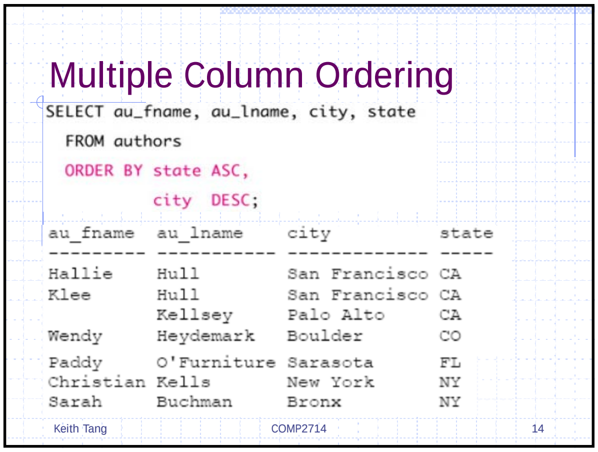
```
au_fname  au_lname    city          state
--------- ----------- ------------- -----
Sarah     Buchman     Bronx         NY
Wendy     Heydemark   Boulder       CO
Hallie    Hull        San Francisco CA
Klee      Hull        San Francisco CA
Christian Kells       New York      NY
          Kellsey     Palo Alto     CA
Paddy     O'Furniture Sarasota      FL
```



Single Column Ordering

```
SELECT au_fname, au_lname, city, state
  FROM authors
  ORDER BY au_fname DESC;
```

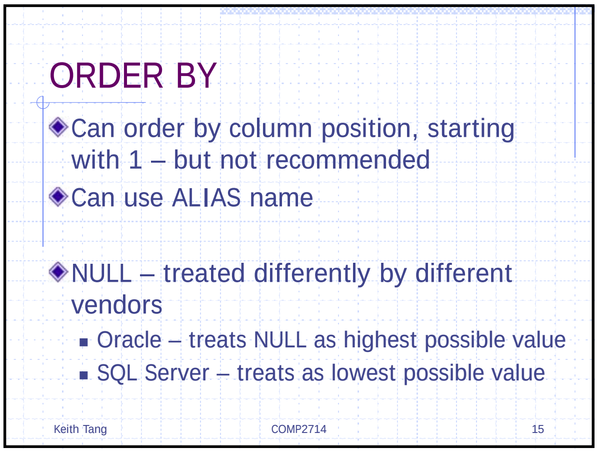
```
au_fname  au_lname    city          state
--------- ----------- ------------- -----
Wendy     Heydemark   Boulder       CO
Sarah     Buchman     Bronx         NY
Paddy     O'Furniture Sarasota      FL
Klee      Hull        San Francisco CA
Hallie    Hull        San Francisco CA
Christian Kells       New York      NY
          Kellsey     Palo Alto     CA
```



Multiple Column Ordering

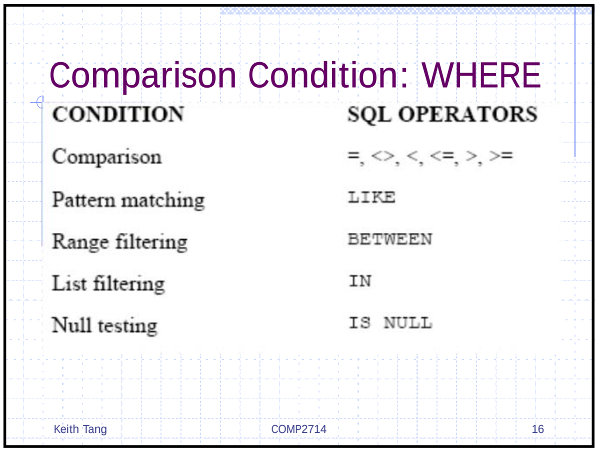
```
SELECT au_fname, au_lname, city, state
  FROM authors
  ORDER BY state ASC,
           city  DESC;
```

```
au_fname  au_lname    city          state
--------- ----------- ------------- -----
Hallie    Hull        San Francisco CA
Klee      Hull        San Francisco CA
          Kellsey     Palo Alto     CA
Wendy     Heydemark   Boulder       CO
Paddy     O'Furniture Sarasota      FL
Christian Kells       New York      NY
Sarah     Buchman     Bronx         NY
```



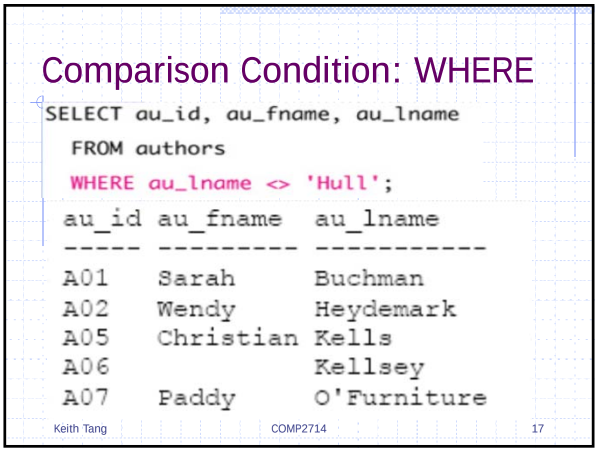
ORDER BY

- Can order by column position, starting with 1 – but not recommended
- Can use ALIAS name
- NULL – treated differently by different vendors
  - Oracle – treats NULL as highest possible value
  - SQL Server – treats as lowest possible value



Comparison Condition: WHERE

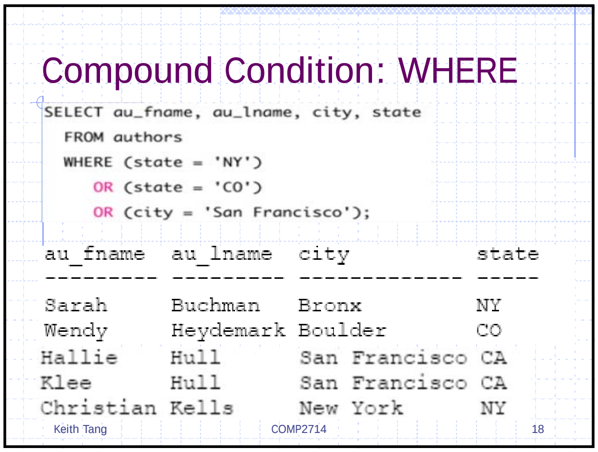
CONDITION	SQL OPERATORS
Comparison	=, <>, <, <=, >, >=
Pattern matching	LIKE
Range filtering	BETWEEN
List filtering	IN
Null testing	IS NULL



Comparison Condition: WHERE

```
SELECT au_id, au_fname, au_lname
  FROM authors
  WHERE au_lname <> 'Hull';
```

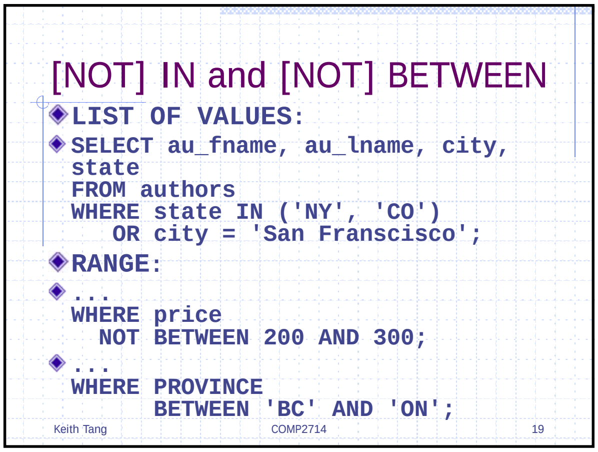
```
 au_id au_fname  au_lname
 ----- --------- -----------
 A01   Sarah     Buchman
 A02   Wendy     Heydemark
 A05   Christian Kells
 A06             Kellsey
 A07   Paddy     O'Furniture
```



Compound Condition: WHERE

```
SELECT au_fname, au_lname, city, state
  FROM authors
  WHERE (state = 'NY')
     OR (state = 'CO')
     OR (city = 'San Francisco');
```

```
au_fname  au_lname  city          state
--------- --------- ------------- -----
Sarah     Buchman   Bronx         NY
Wendy     Heydemark Boulder       CO
Hallie    Hull      San Francisco CA
Klee      Hull      San Francisco CA
Christian Kells     New York      NY
```



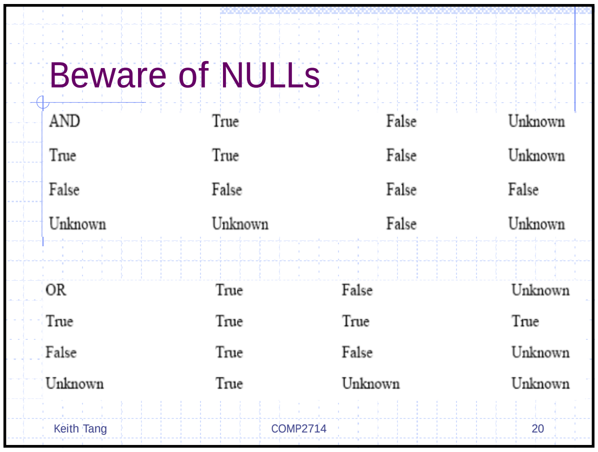
[NOT] IN and [NOT] BETWEEN

- LIST OF VALUES:

-
  ```
  SELECT au_fname, au_lname, city,
  state
  FROM authors
  WHERE state IN ('NY', 'CO')
     OR city = 'San Franscisco';
  ```
- RANGE:

-
  ```
  ...
  WHERE price
    NOT BETWEEN 200 AND 300;
  ```

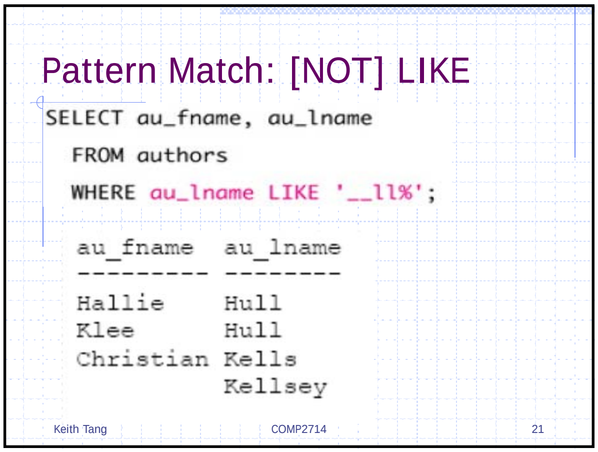
-
  ```
  ...
  WHERE PROVINCE
        BETWEEN 'BC' AND 'ON';
  ```



Beware of NULLs

AND	True	False	Unknown
True	True	False	Unknown
False	False	False	False
Unknown	Unknown	False	Unknown

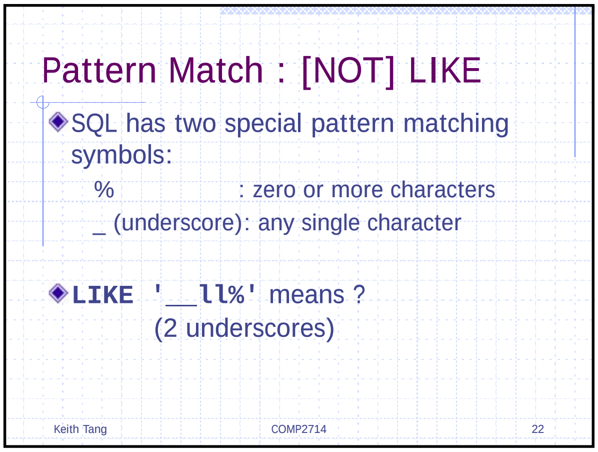
OR	True	False	Unknown
True	True	True	True
False	True	False	Unknown
Unknown	True	Unknown	Unknown



Pattern Match: [NOT] LIKE

```
SELECT au_fname, au_lname
  FROM authors
  WHERE au_lname LIKE '__ll%';
```

```
au_fname  au_lname
--------- --------
Hallie    Hull
Klee      Hull
Christian Kells
          Kellsey
```

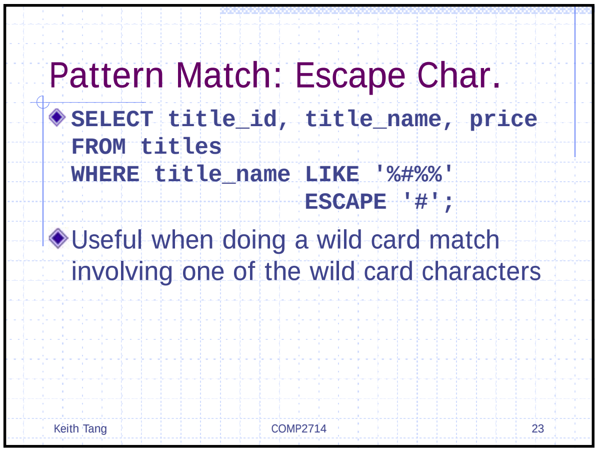


Pattern Match : [NOT] LIKE

- SQL has two special pattern matching symbols:

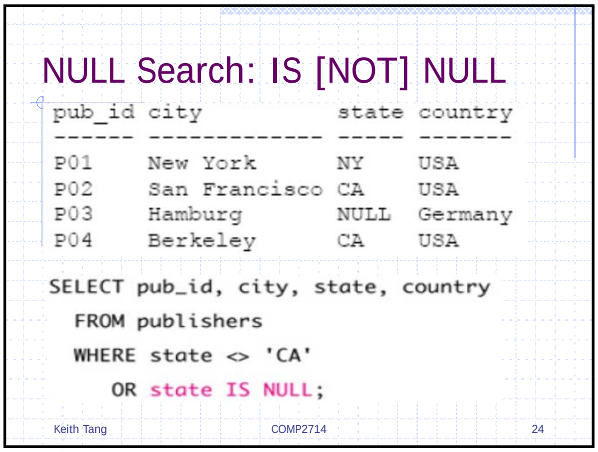
  %   : zero or more characters
  _ (underscore): any single character

- **`LIKE '__ll%'`** means ?
  (2 underscores)



Pattern Match: Escape Char.

-
  ```
  SELECT title_id, title_name, price
  FROM titles
  WHERE title_name LIKE '%#%%'
                  ESCAPE '#';
  ```
- Useful when doing a wild card match involving one of the wild card characters



NULL Search: IS [NOT] NULL

```
pub_id city          state country
------ ------------- ----- -------
P01    New York      NY    USA
P02    San Francisco CA    USA
P03    Hamburg       NULL  Germany
P04    Berkeley      CA    USA
```

```
SELECT pub_id, city, state, country
  FROM publishers
  WHERE state <> 'CA'
     OR state IS NULL;
```