

SQL – Part II

Structured Query Language

LIKE

expr LIKE *pattern*

Patterns

% matches any number of characters

_ matches one character

%

```
SELECT *  
FROM Students  
WHERE FirstName LIKE 'A%'
```

%

```
SELECT *  
FROM Students  
WHERE FirstName LIKE 'An%'
```

%

```
SELECT *  
FROM Students  
WHERE FirstName LIKE 'A%|'
```

—

```
SELECT *  
FROM Students  
WHERE FirstName LIKE '____y'
```

Active learning

Find students:

Area code is 541

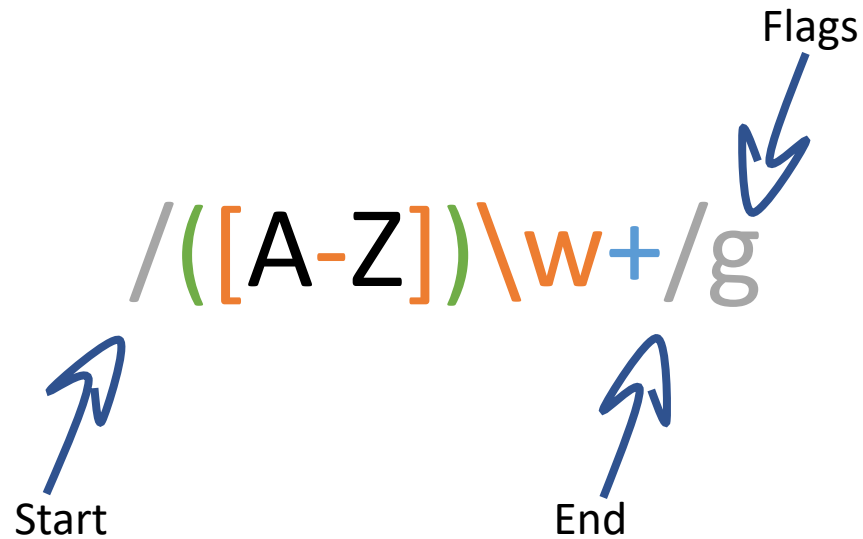
Or Gmail address

REGEXP

expr REGEXP *pattern*

Regular expressions

A regular expression is a sequence of characters that defines a pattern to search through text



The diagram illustrates the components of a regular expression pattern: `/([A-Z])\w+/g`. The pattern is color-coded: the opening parenthesis and closing parenthesis are green, the characters `A-Z` are black, the backslash and `w` are orange, the plus sign is blue, and the final `/g` is grey. Three blue arrows point to specific parts of the pattern: one labeled "Start" points to the first slash, one labeled "End" points to the second slash, and one labeled "Flags" points to the `g` at the end.

Start

End

Flags

Character classes

Character classes	Example	
<code>\d</code> any digit	<code>\d</code>	+1-(444)-555-1234
<code>\D</code> not a digit	<code>\D</code>	+1-(444)-555-1234
<code>\s</code> space	<code>\s</code>	glib jocks vex dwarves!
<code>\S</code> not a space	<code>\S</code>	glib jocks vex dwarves!
<code>\w</code> any character	<code>\w</code>	glib jocks vex dwarves!
<code>\W</code> not a character	<code>\W</code>	glib jocks vex dwarves!
<code>.</code> any character, except <code>\n</code>	<code>.</code>	glib jocks vex dwarves!
<code>[ABC]</code> any character in set	<code>[aeiou]</code>	glib jocks vex dwarves!
<code>[^ABC]</code> negated set	<code>[^aeiou]</code>	glib jocks vex dwarves!
<code>[A-Z]</code> any character in range	<code>[g-s]</code>	abcdefg <hij< h="">klmnopqrstuvwxyz</hij<>

Anchors

Anchors	Example	
^ beginning of string	^\w+	she sells seashells
\$ end of string	\w+\$	she sells seashells
\b word boundary	s\b	she sell s seashell s
\B not word boundary	s\B	s he s ells s ea s hells

Quantifiers and alternation

Quantifiers and alternation		Example	
+	1 or more of the previous	<code>b\w+</code>	b be bee beer beers
*	0 or more of the previous	<code>b\w*</code>	b be bee beer beers
<code>{1,3}</code>	Specified quantity of previous *	<code>b\w{2,3}</code>	b be bee beer beers
?	0 or 1 of the previous (optional)	<code>colou?r</code>	color colour
?	0 or 1 of the previous (lazy)	<code>b\w+?</code>	b be bee beer beers
	or	<code>b(a e i)d</code>	bad bud bod bed bid

* Matches the specified quantity of the previous token. `{1,3}` will match 1 to 3. `{3}` will match exactly 3. `{3,}` will match 3 or more.

Sample text

<http://bit.ly/3bJYpub>

Central Park is an urban park in New York City located between the Upper West and Upper East Sides of Manhattan. It is the fifth-largest park in the city by area, covering 843 acres (341 ha). It is the most visited urban park in the United States with an estimated 38 million visitors annually, and is the most filmed location in the world. [b be bee beer beers] [she sells seashells][+1-212-360-3444]

Following proposals for a large park in Manhattan during the 1840s, it was approved in 1853 to cover 778 acres (315 ha). In 1857, landscape architects Frederick Law Olmsted and Calvert Vaux won a design competition for the park with their "Greensward Plan". Construction began the same year; existing structures, including a majority-Black settlement named Seneca Village, were seized through eminent domain and razed. The park's first areas were opened to the public in late 1858. [bad bud bod bed bid][abcdefghijklmnopqrstuvwxyz][+1 212-310-6600]

There are 21 children's playgrounds in Central Park. The largest, at three acres (12,000 m²), is Heckscher Playground. Central Park includes 36 ornamental bridges, all with different designs. "Rustic" shelters and other structures were originally spread out through the park. Most have been demolished over the years, and several have been restored. The park contains around 9,500 benches in three styles, of which nearly half have small engraved tablets of some kind, installed as part of Central Park's "Adopt-a-Bench" program. These engravings typically contain short personalized messages and can be installed for at least \$10,000 apiece. "Handmade rustic benches" can cost more than half a million dollars and are only granted when the honoree underwrites a major park project.[+1 212 439 6500] [617-826-8977] [617 826 8977] [(617) 826 8977] [(617) 826-8977]

Expression <> JavaScript Flags

/([A-Z])\w+/g

Text Tests NEW 29 matches (0.3ms)

RegExpr was created by gskinner.com, and is proudly hosted by Media Temple.

Edit the Expression & Text to see matches. Roll over matches or the expression for details. PCRE & JavaScript flavors of RegEx are supported. Validate your expression with Tests mode.

The side bar includes a Cheatsheet, full Reference, and Help. You can also Save & Share with the Community, and view patterns you create or favorite in My Patterns.

Explore results with the Tools below. Replace & List output custom results. Details lists capture groups. Explain describes your expression in plain English.

Match last names that have an "a", or an "m", or a "c"

SELECT *

FROM *Students*

WHERE *LastName* **REGEXP** *'a/m/c'*

Match last names that start with "C", or a "D", followed by "a" or "o"

```
SELECT *  
FROM Students  
WHERE LastName REGEXP ' (^C/^D)[ao] '
```

^ start of a string
[] range

Match last names that start with a "D", end with an "h"

SELECT *

FROM *Students*

WHERE *LastName* **REGEXP** ' ^D.*h\$ '

\$ end of a string

* match 0 or more

Match last names ending in "n"

```
SELECT *  
FROM Students  
WHERE LastName REGEXP '.*n$'
```

.ing\$
Match ending in "ing"

Last names starting with D

```
SELECT *  
FROM Students  
WHERE LastName REGEXP '^D\w*'
```

Last names with 5 characters or less

SELECT *

FROM *Students*

WHERE *LastName* **REGEXP** *'^.{1,5}\$'*

last names starting with D and 7 characters or less

SELECT *

FROM *Students*

WHERE *LastName* **REGEXP** *'^D.{1,7}\$'*

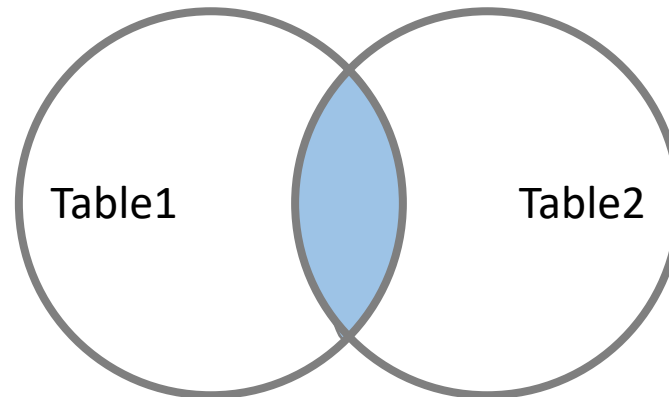
Active learning: Match phone numbers using a regular expression

- Note MySQL syntax differences
- Escaping parenthesis
 - Traditional: `\(`
 - MySQL: `[(`
- Digits
 - Traditional: `\d`
 - MySQL: `[0-9]`
- Space
 - Traditional: `\s`
 - MySQL: `[[:space:]]`

JOINS

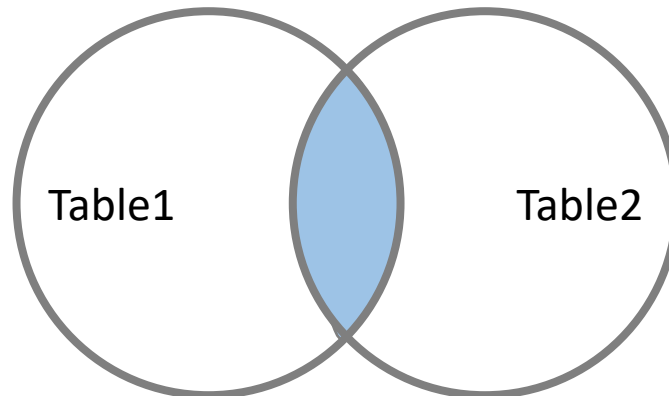
INNER JOIN

INNER JOIN table ON



INNER JOIN

```
SELECT table1.column, table2.column  
FROM table1  
INNER JOIN table2  
ON table1.column = table2.column
```

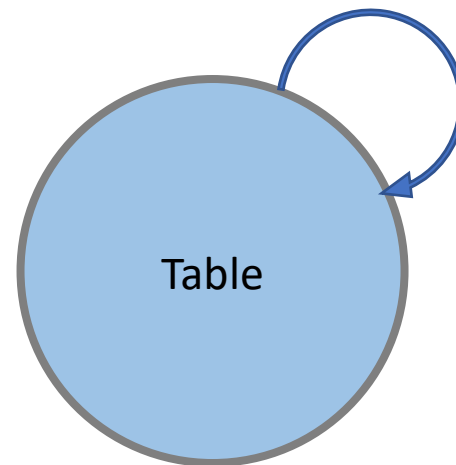


Active learning: where does each student go to college? Write the join.

	StudentID	FirstName	LastName	CollegeID	Name
►	1	Nancy	Davolio	1	MIT
	2	Andrew	Fuller	9	Johns Hopkins
	3	Janet	Leverling	8	Princeton
	4	Margaret	Peacock	3	Dartmouth
	5	Steven	Buchanan	4	Stanford
	6	Michael	Suyama	7	Harvard
	7	Robert	King	6	Columbia
	8	Laura	Callahan	5	Yale
	9	Anne	Dodsworth	2	Brown
	10	Ivy	Johnson	1	MIT
	11	Ana	Trujillo	1	MIT
	12	Thomas	Hardy	9	Johns Hopkins
	13	Antonio	Moreno	5	Yale
	14	Elizabeth	Brown	7	Harvard
	15	Ann	Devon	3	Dartmouth
	16	Ariel	Cruz	2	Brown
	17	Giovanni	Rovelli	6	Columbia
	18	Marie	Bertrand	10	Northwestern
	19	Philip	Cramer	4	Stanford
	20	Michael	Holz	8	Princeton

SELF JOIN

INNER JOIN table ON



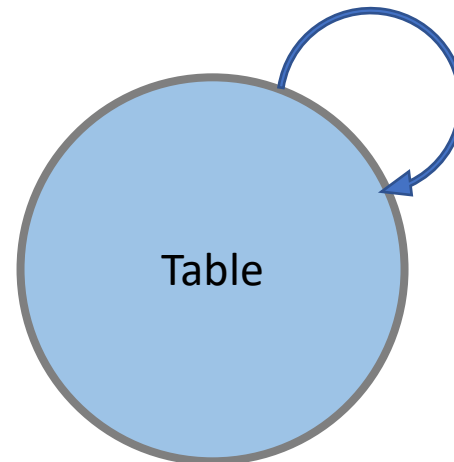
SELF JOIN

SELECT column_names

FROM table **AS** t1

INNER JOIN table **AS** t2

ON t1.column = t2.column



Students	
PK	StudentID
FK	CollegeID
	FriendID
	FirstName
	LastName
	BirthDate
	Email
	Phone
	City
	Region
	Country

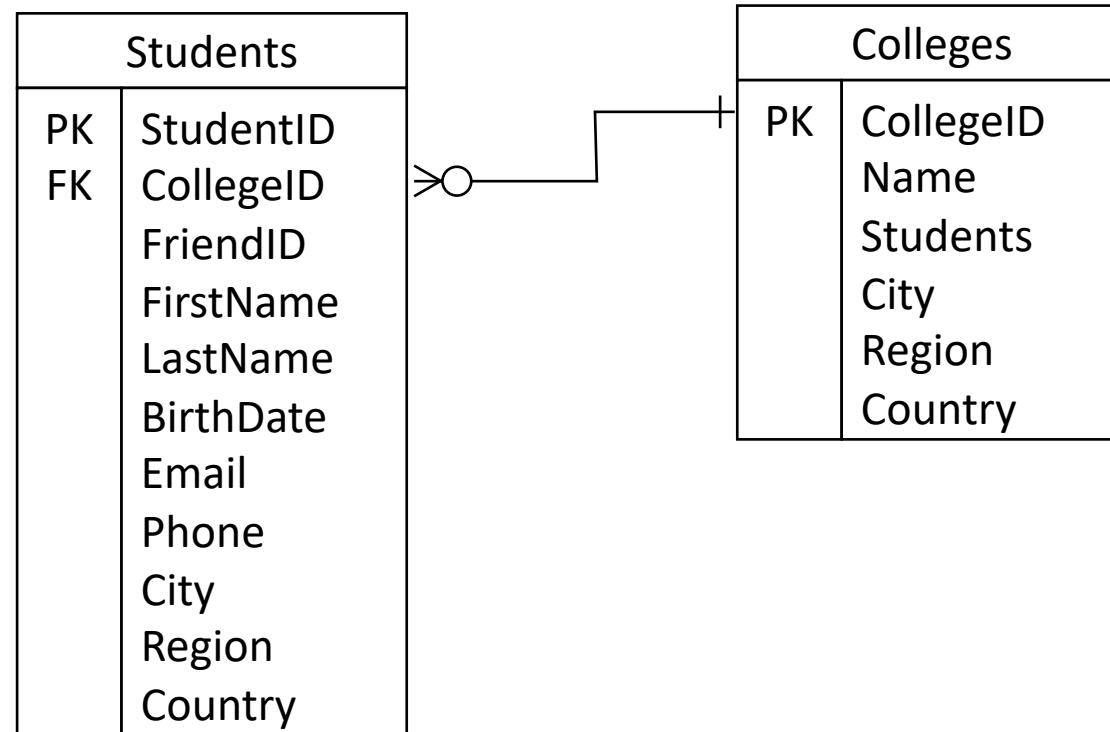
A diagram illustrating a self-join on the Students table. A blue curved arrow originates from the FriendID column and points back to the StudentID column, indicating a relationship between a student and their friend within the same table.

Active learning: what is the name of each friend? Write the join.

	StudentID	FirstName	LastName	FriendID	Buddy
▶	1	Nancy	Davolio	10	Ivy Johnson
	2	Andrew	Fuller	5	Steven Buchanan
	3	Janet	Leverling	1	Nancy Davolio
	4	Margaret	Peacock	9	Anne Dodsworth
	5	Steven	Buchanan	2	Andrew Fuller
	6	Michael	Suyama	8	Laura Callahan
	7	Robert	King	3	Janet Leverling
	8	Laura	Callahan	7	Robert King
	9	Anne	Dodsworth	4	Margaret Peacock
	10	Ivy	Johnson	6	Michael Suyama

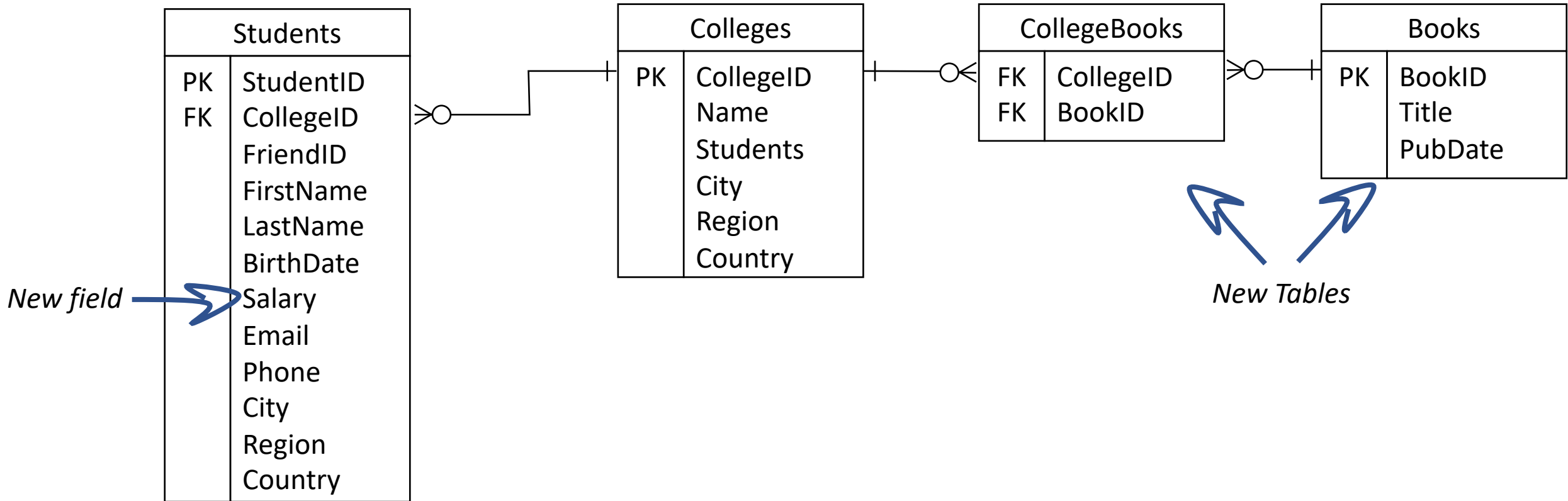
JOINING MULTIPLE TABLES

We started with



Let's add

Colleges can write multiple books.
Books can have multiple colleges.



Multiple table join

Table1 → Table 2, 1:N

Table3 → Table 1, 1:N

SELECT *

FROM table1 t1

JOIN table2 cb

ON t1.columnID = t2.columnID

JOIN table3 t3

ON t3.columnID = t2.columnID



Active learning: colleges for book titles

	BookID	Title	PubDate	CollegeID	BookID	CollegeID	Name	Students	City	Region	Country
►	1	Physics 101	1985-06-21	1	1	1	MIT	11	Cambridge	MA	USA
	2	Math 101	1995-03-15	1	2	1	MIT	11	Cambridge	MA	USA
	3	History 101	1974-05-29	1	3	1	MIT	11	Cambridge	MA	USA
	4	Biology 101	1995-08-10	4	4	4	Stanford	17	Stanford	CA	USA
	5	English 101	2011-09-21	5	5	5	Yale	12	New Haven	CT	USA
	6	Chemistry 101	2015-04-11	6	6	6	Columbia	31	New York	NY	USA
	7	Spanish 101	2008-09-05	7	7	7	Harvard	23	Cambridge	MA	USA
	8	Engr 101	1993-11-23	7	8	7	Harvard	23	Cambridge	MA	USA
	9	Geo 101	1968-12-05	7	9	7	Harvard	23	Cambridge	MA	USA
	10	Biz 101	1979-01-20	1	10	1	MIT	11	Cambridge	MA	USA
	10	Biz 101	1979-01-20	2	10	2	Brown	9	Providence	RI	USA
	10	Biz 101	1979-01-20	3	10	3	Dartmouth	6	Hanover	NH	USA

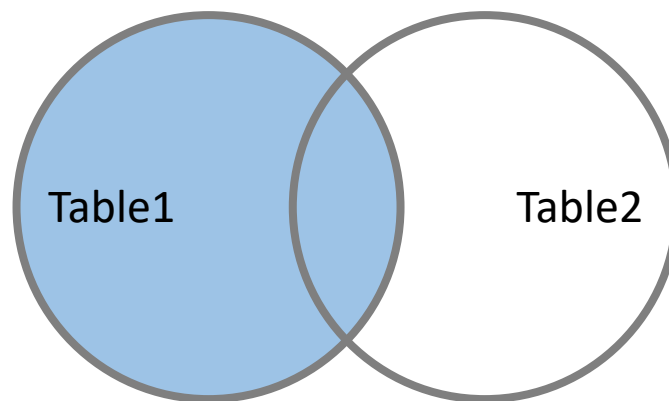
Active learning: books for colleges

	CollegeID	Name	Students	City	Region	Country	CollegeID	BookID	BookID	Title	PubDate
►	1	MIT	11	Cambridge	MA	USA	1	1	1	Physics 101	1985-06-21
	1	MIT	11	Cambridge	MA	USA	1	2	2	Math 101	1995-03-15
	1	MIT	11	Cambridge	MA	USA	1	3	3	History 101	1974-05-29
	4	Stanford	17	Stanford	CA	USA	4	4	4	Biology 101	1995-08-10
	5	Yale	12	New Haven	CT	USA	5	5	5	English 101	2011-09-21
	6	Columbia	31	New York	NY	USA	6	6	6	Chemistry 101	2015-04-11
	7	Harvard	23	Cambridge	MA	USA	7	7	7	Spanish 101	2008-09-05
	7	Harvard	23	Cambridge	MA	USA	7	8	8	Engr 101	1993-11-23
	7	Harvard	23	Cambridge	MA	USA	7	9	9	Geo 101	1968-12-05
	1	MIT	11	Cambridge	MA	USA	1	10	10	Biz 101	1979-01-20
	2	Brown	9	Providence	RI	USA	2	10	10	Biz 101	1979-01-20
	3	Dartmouth	6	Hanover	NH	USA	3	10	10	Biz 101	1979-01-20

MORE JOINS

LEFT JOIN

```
SELECT Students.FirstName, Students.LastName,  
Students.Country, Colleges.Name, Colleges.Country  
FROM Students  
LEFT JOIN Colleges  
ON Students.CollegeID = Colleges.CollegeID;
```

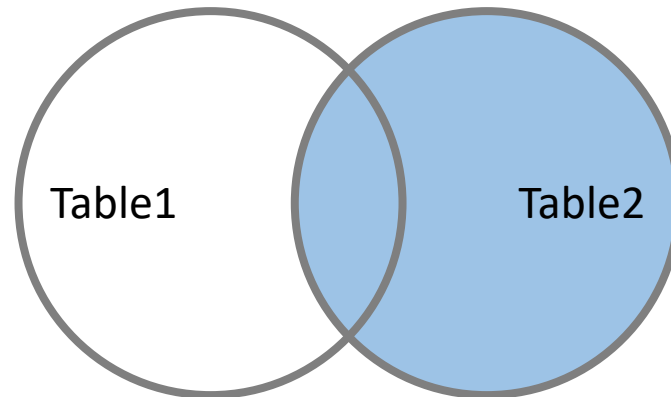


```
SELECT column_names  
FROM table1  
LEFT JOIN table2  
ON table1.column = table2.column
```

FirstName	LastName	Country	Name	Country
▶ Nancy	Davolio	USA	MIT	USA
Andrew	Fuller	USA	Johns Hopkins	USA
Janet	Leverling	USA	Princeton	USA
Margaret	Peacock	USA	Dartmouth	USA
Steven	Buchanan	USA	Stanford	USA
Michael	Suyama	USA	Harvard	USA
Robert	King	USA	Columbia	USA
Laura	Callahan	USA	Yale	USA
Anne	Dodsworth	USA	Brown	USA
Ivy	Johnson	USA	MIT	USA
Ana	Trujillo	USA	MIT	USA
Thomas	Hardy	USA	Johns Hopkins	USA
Antonio	Moreno	USA	Yale	USA
Elizabeth	Brown	USA	Harvard	USA
Ann	Devon	USA	Dartmouth	USA
Ariel	Cruz	USA	Brown	USA
Giovanni	Rovelli	USA	Columbia	USA
Marie	Bertrand	USA	Northwestern	USA
Philip	Cramer	USA	Stanford	USA
Michael	Holz	USA	Princeton	USA

RIGHT JOIN

```
SELECT Students.FirstName, Students.LastName,  
Students.Country, Colleges.Name, Colleges.Country  
FROM Students  
RIGHT JOIN Colleges  
ON Students.CollegeID = Colleges.CollegeID
```

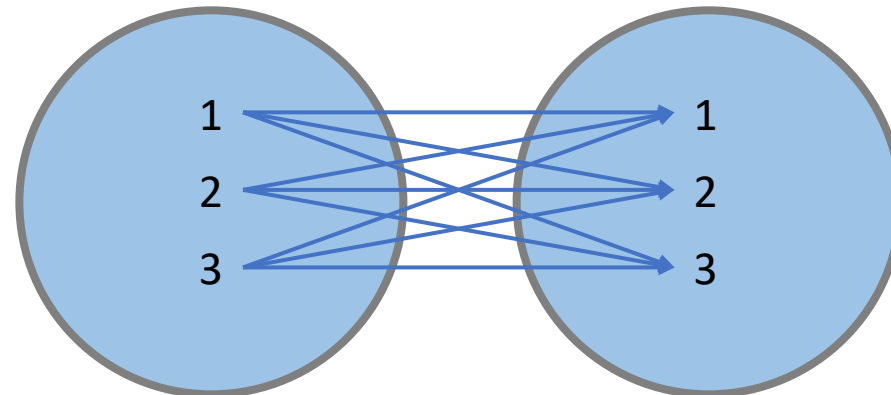


```
SELECT column_names  
FROM table1  
RIGHT JOIN table2  
ON table1.column = table2.column
```

FirstName	LastName	Country	Name	Country
Nancy	Davolio	USA	MIT	USA
Ivy	Johnson	USA	MIT	USA
Ana	Trujillo	USA	MIT	USA
Anne	Dodsworth	USA	Brown	USA
Ariel	Cruz	USA	Brown	USA
Margaret	Peacock	USA	Dartmouth	USA
Ann	Devon	USA	Dartmouth	USA
Steven	Buchanan	USA	Stanford	USA
Philip	Cramer	NULL	Stanford	USA
Laura	Callahan	USA	Yale	USA
Antonio	Moreno	USA	Yale	USA
Robert	King	USA	Columbia	USA
Giovanni	Rovelli	USA	Columbia	USA
Michael	Suyama	USA	Harvard	USA
Elizabeth	Brown	USA	Harvard	USA
Janet	Leverling	USA	Princeton	USA
Michael	Holz	NULL	Princeton	USA
Andrew	Fuller	USA	Johns Hop...	USA
Thomas	Hardy	USA	Johns Hop...	USA
Marie	Bertrand	NULL	Northwest...	USA
NULL	NULL	NULL	Duke	USA
NULL	NULL	NULL	Cornell	USA
NULL	NULL	NULL	Notre Dame	USA
NULL	NULL	NULL	UCLA	USA
NULL	NULL	NULL	Berkeley	USA
NULL	NULL	NULL	Georgetown	USA
NULL	NULL	NULL	Michigan	USA
NULL	NULL	NULL	USC	USA
NULL	NULL	NULL	Tufts	USA
NULL	NULL	NULL	NYU	USA

CROSS JOIN

```
SELECT S.StudentID, S.FirstName, S.LastName,  
C.CollegeID, C.Name  
FROM Students S  
CROSS JOIN Colleges C  
WHERE S.StudentID=1
```



	StudentID	FirstName	LastName	CollegeID	Name
▶	1	Nancy	Davolio	1	MIT
	1	Nancy	Davolio	2	Brown
	1	Nancy	Davolio	3	Dartmouth
	1	Nancy	Davolio	4	Stanford
	1	Nancy	Davolio	5	Yale
	1	Nancy	Davolio	6	Columbia
	1	Nancy	Davolio	7	Harvard
	1	Nancy	Davolio	8	Princeton
	1	Nancy	Davolio	9	Johns Hopkins
	1	Nancy	Davolio	10	Northwestern
	1	Nancy	Davolio	11	Duke
	1	Nancy	Davolio	12	Cornell
	1	Nancy	Davolio	13	Notre Dame
	1	Nancy	Davolio	14	UCLA
	1	Nancy	Davolio	15	Berkeley
	1	Nancy	Davolio	16	Georgetown
	1	Nancy	Davolio	17	Michigan
	1	Nancy	Davolio	18	USC
	1	Nancy	Davolio	19	Tufts
	1	Nancy	Davolio	20	NYU

